

# **Controle de Robôs Móveis por Fusão de Sinais de Controle Usando Filtro de Informação Descentralizado**

**Eduardo Oliveira Freire**

**Tese de Doutorado em Engenharia Elétrica (Automação)**

**Doutorado em Engenharia Elétrica (Automação)**

**Universidade Federal do Espírito Santo**

**Vitória, Agosto de 2002**

# **Controle de Robôs Móveis por Fusão de Sinais de Controle Usando Filtro de Informação Descentralizado**

**Eduardo Oliveira Freire**

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Doutor em Engenharia Elétrica – Automação.

Aprovada em 14/08/2002 por:

---

Prof. Dr. Teodiano Freire Bastos Filho - Orientador, UFES

---

Prof. Dr. Mário Sarcinelli Filho – Orientador, UFES

---

Prof. Dr. Ricardo Carelli – Orientador, UNSJ

---

Prof. Dr. Hansjörg Andreas Schneebeli, UFES

---

Prof. Dr. Oscar Nasisi, UNSJ

---

Prof. Dr. José Leandro Felix Salles, UFES

---

Dr. Marcel Bergerman, Instituto Genius

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória, Agosto de 2002

---

Freire, Eduardo Oliveira, 1972

Controle de Robôs Móveis por Fusão de Sinais de Controle Usando Filtro de Informação Descentralizado. [Vitória] 2002

xv, 143 p., 29,7 cm (UFES, D. Sc., Engenharia Elétrica, 2002)

Tese, Universidade Federal do Espírito Santo, PPGEE.

I. Robótica

I. PPGEE/UFES

II. Título (série)

---

**Dedico este trabalho a Deus, Pai Onipotente,  
a Silvana Cecilia, minha esposa muito amada,  
aos meus pais: Antônio e Rosa, a quem devo a vida,  
aos meus irmãos, Ricardo, Thais, Fernando e Roberto,  
ao meu avô, Antônio Oliveira (in memorian),  
e a toda a minha família, no Brasil e na Argentina.**

# **AGRADECIMENTOS**

Inicialmente, gostaria de agradecer aos meus orientadores, prof. Dr. Teodiano Freire Bastos Filho, prof. Dr. Mário Sarcinelli Filho e prof. Dr. Ricardo Carelli por todo o apoio e experiência que me deram ao longo dos meus estudos de doutorado. Aos professores Teodiano Bastos, Mário Sarcinelli e Hansjörg Schneebeli quero agradecer de forma especial por terem confiado que eu poderia realizar este trabalho, mesmo estando trabalhando em uma universidade a mais de 1500 quilômetros de distância de Vitória. Ao prof. Carelli, desejo agradecer especialmente por todo o apoio que me deu enquanto estava em San Juan – Argentina, que foi indispensável para a realização deste trabalho.

Gostaria de agradecer à FAP-SE e ITP-UNIT por financiarem este projeto e de forma especial à fundação CAPES do Brasil e SETCIP da Argentina pelo suporte que foi dado à cooperação entre a Universidade Federal do Espírito Santo (Brasil) e Universidad Nacional de San Juan (Argentina).

Quero agradecer a todos os colegas do PPGEE/UFES que me ajudaram enquanto realizávamos trabalhos em conjunto nas disciplinas necessárias para a obtenção dos créditos, sobretudo no desenvolvimento do robô móvel Brutus, dos quais, por precaução, eu prefiro não citar nenhum nome, evitando assim ser injusto por esquecer algum deles.

Desejo também agradecer a todos colegas, professores e funcionários do INAUT-UNSJ que me receberam tão bem, permitindo que eu me sentisse em casa mesmo em outro país, e por toda ajuda que me prestaram enquanto estive lá.

Agradeço também a todos os colegas da UNIT, à PAPGP, ao ITP, à direção do CCFT, aos coordenadores dos cursos de Ciência da Computação e Sistemas de Informação e aos que foram meus alunos nesta fase de minha vida acadêmica.

Quero agradecer a todas as pessoas, em Aracaju, Vitória e San Juan, que ao longo desses anos me apoiaram no meu trabalho.

Finalmente, desejo agradecer à minha esposa, Silvana Cecilia, aos meus pais, Antônio e Rosa, ao meu avô, Antônio Oliveira (in memorian), aos meus irmãos, Ricardo, Thais e Fernando, à minha sogra e meu sogro, D. Maria e Don Angel, ao meu cunhado Roberto, sua esposa Graciela, seus filhos, Soledad, Marianela e Alejandro, e a toda a minha família, no Brasil e na Argentina, por todo o amor, compreensão, paciência, apoio e carinho que formaram a base, que sob a benção de Deus e a interseção de Nossa Senhora, me permitiram atingir este objetivo tão importante em minha vida.

# SUMÁRIO

## **Capítulo 1**

<b>Introdução</b>	<b>16</b>
1.1 Introdução	16
1.2 Trabalhos Relacionados	17
1.2.1 Arquitetura AuRA [47]	17
1.2.2 Arquitetura DAMN [30][32]	18
1.2.3 Abordagem Dinâmica [11][21][36]	19
1.2.4 Multivaluated Logic Approach [3][4][5][6][7][51]	20
1.2.5 Multiple Objective Behavior-Based Control [38][39]	21
1.2.6 Controle Concorrente [9]	22
1.3 Definição do Problema	23
1.4 Objetivo	24
1.4.1 Tarefas	24
1.5 Contribuições desta Tese de Doutorado	25
1.6 Estrutura da Tese	26

## **Capítulo 2**

<b>O Filtro de Informação Descentralizado</b>	<b>28</b>
2.1 O Filtro de Kalman	29
2.2 O Filtro de Kalman Descentralizado	31
2.3 O Filtro de Informação [1]	32
2.4 O Filtro de Informação Descentralizado	33
2.5 Comparação entre o Filtro de Informação e o Filtro de Kalman	34
2.6 Filtro de Kalman, Filtro de Kalman Descentralizado, Filtro de Informação e Filtro de Informação Descentralizado: Um Estudo de Caso	35

## **Capítulo 3**

<b>A Arquitetura de Controle Proposta</b>	<b>41</b>
3.1 Interpretação das Covariâncias	43
3.2 O Cálculo das Covariâncias	44

## **Capítulo 4**

<b>Análise de Estabilidade</b>	<b>47</b>
4.1 Fusão das Saídas de Vários Controladores com o Mesmo Objetivo de Controle	48
4.2 Fusão das Saídas de Vários Controladores com Objetivos de Controle Diferentes	52
4.3 Sistema Supervisor	53
4.3.1 Determinação dos Controladores que Estão Fora de Contexto	55
4.3.2 Determinação de Quais Controladores Devem Ser Eliminados do Processo de Fusão	56
4.3.3 Determinação da Ocorrência de Mudança de Fase de Navegação	57

<b>Capítulo 5</b>	
<b>Controladores Utilizados</b>	<b>59</b>
<b>5.1 Controlador para Seguir Corredores</b>	<b>59</b>
5.1.1 Equações Cinemáticas	60
5.1.2 Projeto do Controlador	61
5.1.3 Limitação dos Sinais de Controle	63
5.1.4 Estimação das Variáveis de Estado	63
5.1.5 Simulações e Resultados Experimentais	70
<b>5.2 Controlador para Seguir Paredes</b>	<b>74</b>
<b>5.3 Controlador de Posição Final</b>	<b>77</b>
<b>5.4 Controlador para Evitar Obstáculos</b>	<b>78</b>
 <b>Capítulo 6</b>	
<b>Simulações e Resultados Experimentais</b>	<b>81</b>
<b>6.1 O Robô Móvel PIONEER 2DX</b>	<b>81</b>
<b>6.2 Simulações e Experimentos Realizados</b>	<b>82</b>
6.2.1 Teste #1	83
6.2.2 Teste #2	95
6.2.3 Teste #3	107
6.2.4 Teste #4	119
<b>6.3 Análise dos Resultados</b>	<b>131</b>
 <b>Capítulo 7</b>	
<b>Conclusões</b>	<b>132</b>
 <b>Bibliografia</b>	<b>135</b>
<b>Bibliografia Referenciada</b>	<b>135</b>
<b>Bibliografia Consultada</b>	<b>140</b>

## **LISTA DE TABELAS**

<i>Tabela 1</i>	<i>Média e desvio padrão dos dados medidos pelos quatro sensores, dos dados resultantes dos quatro filtros em questão e da média instantânea das medidas dos quatro sensores.</i>	<i>36</i>
<i>Tabela 2</i>	<i>Correlações cruzadas entre os dados obtidos pelos filtros analisados quando tomados dois a dois.</i>	<i>39</i>
<i>Tabela 3</i>	<i>Número de FLOPS resultante quando se utilizam os filtros FK, FKD, FI e FID</i>	<i>40</i>
<i>Tabela 4</i>	<i>Regras fuzzy utilizadas para determinar a covariância do controlador responsável por executar a tarefa de navegar por corredores.</i>	<i>45</i>
<i>Tabela 5</i>	<i>Regras fuzzys utilizadas para determinar as covariâncias dos controladores responsáveis por executar as tarefas de buscar a posição final (R2), evitar obstáculos (R3) e seguir paredes (R4).</i>	<i>46</i>
<i>Tabela 6</i>	<i>Regras fuzzy utilizadas para determinar as covariâncias associadas aos filtros locais.</i>	<i>66</i>
<i>Tabela 7</i>	<i>Índices de Avaliação de Desempenho do Teste #1.</i>	<i>83</i>
<i>Tabela 8</i>	<i>Índices de Avaliação de Desempenho do Teste #2.</i>	<i>95</i>
<i>Tabela 9</i>	<i>Índices de Avaliação de Desempenho do Teste #3.</i>	<i>107</i>
<i>Tabela 10</i>	<i>Índices de Avaliação de Desempenho do Teste #4.</i>	<i>119</i>



# LISTA DE ILUSTRAÇÕES

Figura 1	Taxonomia dos ASM's [37].	17
Figura 2	Comparação do resultado da fusão dos dados provenientes de dois sensores, usando Filtro de Kalman, com a simples média aritmética instantânea dos mesmos.	28
Figura 3	O Filtro de Kalman [45].	31
Figura 4	Exemplo de Filtro de Kalman Descentralizado com dois filtros locais.	32
Figura 5	Funções de densidade de probabilidade da média instantânea das medidas dos sensores, do Filtro de Kalman, do Filtro de Informação Descentralizado e dos quatro sensores de ultra-som.	37
Figura 6	Dados medidos pelos quatro sensores.	38
Figura 7	Média instantânea dos dados obtidos pelos quatro sensores.	38
Figura 8	Resultados obtidos pelos filtros FK, FKD, FI e FID.	39
Figura 9	Implementação da arquitetura proposta para o controle da navegação de um robô móvel em um ambiente composto por corredores e obstáculos.	42
Figura 10	Funções de pertinência das variáveis de entrada (antecedentes).	45
Figura 11	Sistema com somente um controlador.	48
Figura 12	Fusão das saídas de diferentes controladores.	50
Figura 13	Sistema Supervisor implementado para garantir o cumprimento da conjectura para que o sistema apresente um "bom comportamento".	54
Figura 14	Posição e orientação do robô.	60
Figura 15	O robô no corredor.	61
Figura 16	Variáveis de estado $\tilde{x}(t)$ e $\phi(t)$ utilizadas pelo controlador projetado para seguir corredores.	64
Figura 17	Processo de estimação das variáveis de estado $\tilde{x}(t)$ e $\phi(t)$ .	65
Figura 18	Função de pertinência da variável de fuzzy de entrada dif.	65
Figura 19	Função de pertinência da variável de fuzzy de entrada $d_{dir} \times d_{esq}$ .	66
Figura 20	Variável de estado $\tilde{x}(t)$ calculada utilizando os dados odométricos corrigidos (acima) e utilizando os dados fornecidos pelo sistema de sensoramento ultra-sônico (abaixo).	68
Figura 21	Sinal resultante do processo de fusão (acima), e a estimativa da variável $\tilde{x}(t)$ (abaixo).	69
Figura 22	Variável de estado $\phi(t)$ calculada utilizando os dados odométricos corrigidos (acima) e utilizando os dados fornecidos pelo sistema de sensoramento ultra-sônico (abaixo).	69
Figura 23	Sinal resultante do processo de fusão (acima), e a estimativa da variável $\phi(t)$ (abaixo).	70
Figura 24	Simulação do controlador projetado para executar a tarefa de seguir corredores.	71
Figura 25	Estrutura utilizada para incorporar o controlador de evitar obstáculos como uma perturbação ao sistema de controle.	71
Figura 26	Caminho percorrido pelo robô em um dos experimentos realizados para demonstrar o desempenho do controlador projetado para seguir corredores, com a capacidade de evitar obstáculos (considerada como uma perturbação).	72

Figura 27	Variáveis de estado $\tilde{x}(t)$ (acima) e $\varphi(t)$ (abaixo) estimadas.	72
Figura 28	Velocidades angulares produzidas pelo controlador projetado para seguir corredores (acima) e velocidade angular medida pelo robô móvel (abaixo).	73
Figura 29	Variáveis de estado para o controlador de seguir paredes.	74
Figura 30	Caminho percorrido pelo robô.	75
Figura 31	Variáveis de estado $\tilde{d}$ (acima) e $\varphi(t)$ (abaixo) estimadas.	76
Figura 32	Velocidades angulares produzidas pelo controlador projetado para seguir paredes (acima) e velocidade angular medida pelo robô móvel (abaixo).	76
Figura 33	A força fictícia $F$ .	78
Figura 34	Diagrama de blocos do sistema de controle baseado em impedância [24][49].	79
Figura 35	Robô móvel PIONEER 2DX.	82
Figura 36	Teste #1.	83
Figura 37	Covariâncias associadas aos controladores (Simulação #1).	84
Figura 38	Covariâncias associadas aos controladores sobrepostas (Simulação #1).	84
Figura 39	Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #1).	85
Figura 40	Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #1).	85
Figura 41	Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #1).	86
Figura 42	Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Simulação #1).	86
Figura 43	Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #1).	87
Figura 44	Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #1).	87
Figura 45	Função energia do sistema (Simulação #1).	88
Figura 46	Estados dos controladores (Simulação #1).	88
Figura 47	Caminho percorrido pelo robô (Simulação #1).	89
Figura 48	Covariâncias associadas aos controladores (Experimento #1).	89
Figura 49	Covariâncias associadas aos controladores sobrepostas (Experimento #1).	90
Figura 50	Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento #1).	90
Figura 51	Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento #1).	91
Figura 52	Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #1).	91
Figura 53	Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #1).	92
Figura 54	Funções energia associadas aos quatro controladores envolvidos no processo (Experimento #1).	92
Figura 55	Função energia do sistema e indicações de mudanças de fase de navegação (Experimento #1).	93
Figura 56	Função energia do sistema (Experimento #1).	93
Figura 57	Estados dos controladores (Experimento #1).	94

Figura 58	Caminho percorrido pelo robô (Experimento #1).	94
Figura 59	Teste #2.	95
Figura 60	Covariâncias associadas aos controladores (Simulação #2).	96
Figura 61	Covariâncias associadas aos controladores sobrepostas (Simulação #2).	96
Figura 62	Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #2).	97
Figura 63	Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #2).	97
Figura 64	Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #2).	98
Figura 65	Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Simulação #2).	98
Figura 66	Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #2).	99
Figura 67	Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #2).	99
Figura 68	Função energia do sistema (Simulação #2).	100
Figura 69	Estados dos controladores (Simulação #2).	100
Figura 70	Caminho percorrido pelo robô (Simulação #2).	101
Figura 71	Covariâncias associadas aos controladores (Experimento 2).	101
Figura 72	Covariâncias associadas aos controladores sobrepostas (Experimento 2).	102
Figura 73	Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento 2).	102
Figura 74	Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento 2).	103
Figura 75	Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #2).	103
Figura 76	Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #2).	104
Figura 77	Funções energia associadas aos quatro controladores envolvidos no processo (Experimento 2).	104
Figura 78	Função energia do sistema e indicações de mudanças de fase de navegação (Experimento 2).	105
Figura 79	Função energia do sistema (Experimento 2).	105
Figura 80	Estados dos controladores (Experimento 2).	106
Figura 81	Caminho percorrido pelo robô (Experimento 2).	106
Figura 82	Teste #3.	107
Figura 83	Covariâncias associadas aos controladores (Simulação #3).	108
Figura 84	Covariâncias associadas aos controladores sobrepostas (Simulação #3).	108
Figura 85	Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #3).	109
Figura 86	Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #3).	109

Figura 87	Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #3).	110
Figura 88	Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Simulação #3).	110
Figura 89	Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #3).	111
Figura 90	Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #3).	111
Figura 91	Função energia do sistema (Simulação #3).	112
Figura 92	Estados dos controladores (Simulação #3).	112
Figura 93	Caminho percorrido pelo robô (Simulação #3).	113
Figura 94	Covariâncias associadas aos controladores (Experimento #3).	113
Figura 95	Covariâncias associadas aos controladores sobrepostas (Experimento #3).	114
Figura 96	Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento #3).	114
Figura 97	Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento #3).	115
Figura 98	Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #3).	115
Figura 99	Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #3).	116
Figura 100	Funções energia associadas aos quatro controladores envolvidos no processo (Experimento #3).	116
Figura 101	Função energia do sistema e indicações de mudanças de fase de navegação (Experimento #3).	117
Figura 102	Função energia do sistema (Experimento #3).	117
Figura 103	Estados dos controladores (Experimento #3).	118
Figura 104	Caminho percorrido pelo robô (Experimento #3).	118
Figura 105	Teste #4	119
Figura 106	Covariâncias associadas aos controladores (Simulação #4).	120
Figura 107	Covariâncias associadas aos controladores sobrepostas (Simulação #4).	120
Figura 108	Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #4).	121
Figura 109	Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #4).	121
Figura 110	Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #4).	122
Figura 111	Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (simulação #4).	122
Figura 112	Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #4).	123
Figura 113	Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #4).	123
Figura 114	Função energia do sistema (Simulação #4).	124
Figura 115	Estados dos controladores (Simulação #4).	124

<i>Figura 116</i>	<i>Caminho percorrido pelo robô (Simulação #4).</i>	<i>125</i>
<i>Figura 117</i>	<i>Covariâncias associadas aos controladores (Experimento #4).</i>	<i>125</i>
<i>Figura 118</i>	<i>Covariâncias associadas aos controladores sobrepostas (Experimento #4).</i>	<i>126</i>
<i>Figura 119</i>	<i>Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento #4).</i>	<i>126</i>
<i>Figura 120</i>	<i>Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento #4).</i>	<i>127</i>
<i>Figura 121</i>	<i>Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #4).</i>	<i>127</i>
<i>Figura 122</i>	<i>Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #4).</i>	<i>128</i>
<i>Figura 123</i>	<i>Funções energia associadas aos quatro controladores envolvidos no processo (Experimento #4).</i>	<i>128</i>
<i>Figura 124</i>	<i>Função energia do sistema e indicações de mudanças de fase de navegação (Experimento #4).</i>	<i>129</i>
<i>Figura 125</i>	<i>Função energia do sistema (Experimento #4).</i>	<i>129</i>
<i>Figura 126</i>	<i>Estados dos controladores (Experimento #4).</i>	<i>130</i>
<i>Figura 127</i>	<i>Caminho percorrido pelo robô (Experimento #4).</i>	<i>130</i>

## **RESUMO**


Esta Tese de Doutorado propõe uma abordagem alternativa para lidar com o problema da navegação de robôs móveis. Esta nova abordagem foi denominada *fusão de sinais de controle*. A técnica proposta apresentou bons resultados ao ser utilizada para controlar um robô na execução de tarefas relativamente complexas, como é demonstrado nos experimentos apresentados. A técnica é desenvolvida com base no Filtro de Informação Descentralizado, que é derivado aqui a partir das equações do Filtro de Kalman Descentralizado e do Filtro de Informação. Controladores de movimento disponíveis na literatura, e outros introduzidos aqui pela primeira vez, são utilizados para gerar sinais de controle. Estes sinais são fusionados utilizando um Filtro de Informação Descentralizado para produzir o sinal de saída o qual é enviado aos atuadores do robô. Também é realizada uma análise de estabilidade da arquitetura de controle proposta.

# **ABSTRACT**

This thesis proposes an alternative approach to deal with the problem of mobile robot navigation, which is called *fusion of control signals*. The proposed technique has presented good results when a robot has to execute relatively complex tasks, like it is shown in the experiments here presented. The technique is developed with basis on the decentralised information filter, whose equations are here derived from the equations of the decentralised Kalman filter and the information filter. Motion controllers available in the literature and others introduced here for the first time are used to produce the control signals that are fused using a decentralised information filter in order to produce the overall output signal to be sent to the actuators. A stability analysis of the proposed architecture is also addressed.

# Capítulo 1

## Introdução



O presente capítulo tem por objetivo introduzir uma nova arquitetura para o controle da navegação de robôs móveis, arquitetura esta que é a contribuição mais importante da presente Tese de Doutorado. A fim de contextualizar a contribuição contida na Tese, o texto inicia com uma análise sucinta das arquiteturas de controle mais representativas que atualmente são utilizadas para realizar tal controle de navegação. Na sequência são apresentados os objetivos do trabalho e destacadas as principais contribuições da Tese. Finalmente, o capítulo se encerra apresentando a estrutura do texto global.

### 1.1 Introdução

Um importante aspecto no controle de robôs móveis consiste em decidir qual ação deve ser executada no instante seguinte. Este tipo de problema é conhecido como *Action Selection Problem* – ASP, ou *Behaviour Co-ordination Problem* [39]. Esquemas ou arquiteturas de controle que são utilizadas para solucionar este tipo de problema são conhecidas como ASM's (*Action Selection Mechanisms*), que podem ser agrupados em duas categorias: esquemas baseados em arbitração e esquemas baseados em fusão de comandos, conforme ilustra a Figura 1.



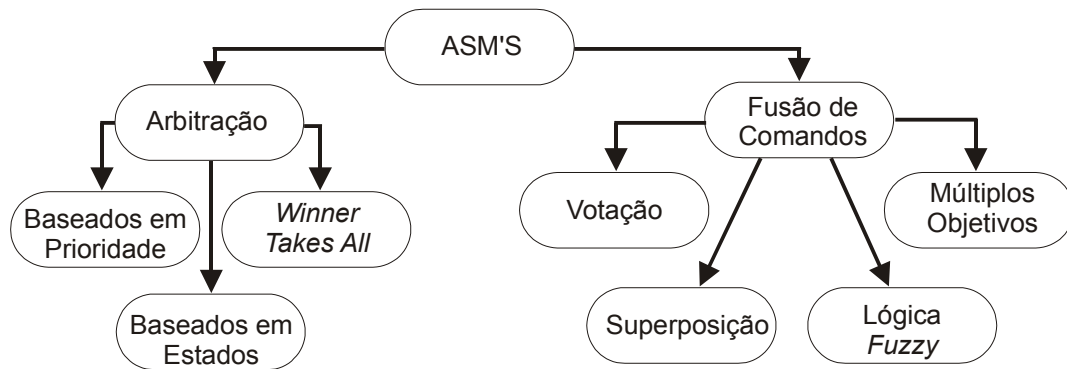


Figura 1 Taxonomia dos ASM's [37].

Nos esquemas baseados em arbitração, a cada momento um único controlador tem o controle total do sistema. Estes esquemas estão subdivididos em três categorias: mecanismos baseados em prioridades, mecanismos tipo *Winner-Takes-All* e mecanismos baseados em estados. Exemplos de esquemas baseados em arbitração incluem a arquitetura de Subsunção de Brooks [42], Sistemas a Eventos Discretos e Redes de Ativação [39].

Por sua vez, os esquemas baseados em fusão de comandos permitem que um conjunto de controladores compartilhe o controle do sistema em cada momento. Os esquemas baseados em fusão de comandos podem ser divididos em quatro categorias: Votação (por exemplo DAMN [32]), Superposição (por exemplo AuRA [47]), Múltiplos Objetivos (por exemplo *Multiple Objective Decision-Making Control* [39]) e Lógica Fuzzy (por exemplo *Multivaluated Logic Approach* [7]) Outro exemplo de um ASM baseado na fusão de comandos é a Abordagem Dinâmica para robôs baseados em Comportamentos<sup>1</sup> [11][21].

A abordagem aqui proposta se constitui em um esquema de fusão de comandos, como se pode ver em detalhes no Capítulo 3.

## 1.2 Trabalhos Relacionados

Existem outros trabalhos que, como a presente proposta, propõem algum tipo de fusão das saídas de distintos controladores ou Comportamentos. A seguir é realizada uma breve análise de alguns dos mais significativos deles.

### 1.2.1 Arquitetura AuRA [47]

A arquitetura AuRA (*Autonomous Robot Architecture*), é uma arquitetura híbrida, composta de um sistema hierárquico, responsável pelo planejamento deliberativo, e um sistema reativo responsável por tarefas de baixo nível (como por exemplo desviar de obstáculos), conhecido como Controlador de Esquemas. Este controlador de esquemas é um sistema reativo com um tipo de controle diferente dos seus antecessores, como a arquitetura

<sup>1</sup> Ao longo do texto, a palavra “comportamento” pode ter duas conotações: pode significar o resultado de uma sequência de ações ao longo do tempo, ou pode se referir a um módulo de tomada de decisões que é projetado para alcançar um determinado objetivo. Para facilitar a diferenciação, no primeiro caso, a palavra será grafada com a primeira letra minúscula, e no segundo, com a primeira letra maiúscula.

Subsunção de Brooks [42] por exemplo, pelo fato de que não é necessário arbitrar entre os Comportamentos e não existem camadas de Comportamentos. Por outro lado, nos sistemas reativos, para cada contexto do ambiente de operação, o sistema deliberativo define quais dos Comportamentos disponíveis devem ser utilizados para a execução da tarefa desejada.

Os esquemas são codificados utilizando-se uma analogia com o método dos campos potenciais [27]. Porém, os esquemas são muito mais simples do ponto de vista computacional do que o método clássico dos campos potenciais, pois o campo potencial só é calculado na posição do robô, e não em todos os pontos do espaço [46]. Isto permite que os esquemas sejam executados em tempo real, que é um requisito básico para um sistema reativo. Entretanto, como os métodos baseados em campos potenciais são conhecidos por apresentarem problemas com mínimos locais e problemas cíclicos [32][46][47], justifica-se o uso do planejamento deliberativo em conjunto com o controle reativo.

Nesta arquitetura, a cada esquema motor está associado um esquema sensor. A percepção é conduzida no sentido de saber o que é necessário: cada esquema sensor provê a informação necessária para cada Comportamento reagir e as respostas de cada esquema motor aos estímulos recebidos acontecem na forma de vetores. Os esquemas motores podem operar assincronamente, produzindo respostas no menor tempo possível. Para solucionar cada tarefa, alguns Comportamentos são agrupados de maneira adequada. A importância relativa de cada Comportamento é codificada através de um ganho e os vetores produzidos como saídas de cada Comportamento (ou esquema) são multiplicados pelos ganhos associados e são então adicionados vetorialmente para obter a reação do robô ao ambiente de operação. Os ganhos podem ser alterados por um sistema de planejamento ou por um operador humano.

Em resumo, AuRA é uma arquitetura assíncrona que apresenta codificação contínua de respostas, utilizando uma analogia com o método dos campos potenciais. Nesta arquitetura, o método de coordenação entre os diversos Comportamentos é cooperativo, através de soma vetorial ponderada [46].

## 1.2.2 Arquitetura DAMN [30][32]

A arquitetura DAMN (*Distributed Architecture for Mobile Navigation*) é uma arquitetura reativa, onde vários Comportamentos, operando de forma independente e assíncrona, determinam de forma cooperativa a ação do robô.

Esta arquitetura consiste em um grupo de Comportamentos distribuídos e assíncronos que se comunicam com um árbitro centralizado através do envio de votos a favor daquelas ações que satisfaçam seus objetivos e contra as que não satisfaçam. Um árbitro, então, é responsável por executar a fusão dos votos dos Comportamentos e gerar as ações que melhor reflitam os objetivos prioritários do sistema. A cada Comportamento está associado um peso, o que permite estabelecer um esquema de prioridade entre os Comportamentos. Podem existir múltiplos árbitros operando paralelamente para o controle de velocidade linear e angular, o que é uma característica da arquitetura DAMN [46]. O uso de controle compartilhado e distribuído permite que múltiplos níveis de planejamento sejam utilizados para a tomada de decisões sem a necessidade de uma estrutura hierárquica. Além disso, a natureza distribuída e assíncrona da arquitetura permite que diversos objetivos e limitações sejam levados em consideração ao mesmo tempo. Assim, a arquitetura DAMN provê um Comportamento

orientado a objetivo que é coerente e racional, ao mesmo tempo em que preserva a capacidade de resposta em tempo real ao ambiente de operação.

Em resumo, DAMN é uma arquitetura assíncrona que apresenta codificação discreta de resposta, através de conjuntos de votos, e um método de coordenação entre os Comportamentos que consiste em múltiplos árbitros independentes do tipo *Winner-Takes-All* [46].

### 1.2.3 Abordagem Dinâmica [11][21][36]

A abordagem dinâmica fornece vários conceitos e princípios que são baseados na teoria matemática dos sistemas dinâmicos e em elementos de redes neurais. Estes conceitos podem ser utilizados como uma linguagem teórica que permite o desenvolvimento de arquiteturas de controle para a navegação de robôs móveis. As principais idéias desta linguagem são:

- (a) O conceito de variáveis comportamentais, que consistem em variáveis que podem descrever, parametrizar e representar internamente um certo comportamento (estado do sistema). Estas variáveis definem as dimensões comportamentais ao longo das quais cada Comportamento pode variar. Elas devem ser escolhidas de modo que a cada instante um Comportamento deve estar associado com valores particulares de suas variáveis comportamentais, e os requerimentos de uma tarefa sejam expressos como valores ou conjuntos de valores destas variáveis.
- (b) O conceito de dinâmica comportamental, de acordo com o qual os Comportamentos são gerados como soluções atratoras de sistemas dinâmicos.
- (c) A dinâmica de campos neurais estende os princípios anteriores ao conceito de representação neural da informação.

Os cursos de tempo das variáveis comportamentais são obtidos como soluções de atratores para sistemas dinâmicos (dinâmica comportamental), formuladas para expressar os requisitos do sistema como forças atrativas ou repulsivas.

Através da escolha das variáveis e ajuste nas escalas de tempo o sistema deve ser sintonizado para estar sempre em um atrator, ou ao menos perto de um. Informações sensoriais ou informações de outros módulos comportamentais (sistemas dinâmicos ou outras variáveis comportamentais) determinam a localização, força e alcance das contribuições atrativas e repulsivas da dinâmica comportamental.

A interação entre múltiplas contribuições para a dinâmica comportamental (como fontes de informação sensorial) pode resultar em uma cooperação ou competição, o que pode levar a uma mudança de comportamento ou simplesmente a um ajuste comportamental.

A abordagem dinâmica aplicada ao controle de navegação de robôs móveis é uma arquitetura síncrona, a qual apresenta codificação contínua de resposta, através de campos vetoriais. Por outro lado, o método de coordenação entre os Comportamentos consiste na soma dos campos vetoriais produzidos por cada Comportamento.

### 1.2.4 *Multivaluated Logic Approach* [3][4][5][6][7][51]

Esta arquitetura é composta de 3 camadas: Esquemas de Controle, Esquemas Comportamento e Planificadores, que interagem com o robô assim como com o ambiente em distintos níveis de abstração: sinal/estímulo, intermediário (símbolo-para-sinal) e simbólico, respectivamente [37].

Os Esquemas de Controle descrevem tipos de movimentos baseados no estado interno do robô e nos dados fornecidos pelos sensores. Os planificadores, no outro extremo, sintetizam planos que são baseados em descrições simbólicas. Os esquemas Comportamentos preenchem o espaço entre os planificadores e os esquemas controladores, contendo partes tanto do nível de abstração simbólico quanto de estímulo.

Os esquemas de controle são definidos como um mapeamento de um conjunto  $S$  de estados para preferências em um conjunto  $A$  de ações a executar. Formalmente, o mapeamento  $D: S \times A \rightarrow [0,1]$ , mede o *grau de desejo*  $D(s,a)$  em executar a ação  $a$  na situação  $s$ . Na prática, cada esquema de controle é implementado por um conjunto de regras *fuzzy* do tipo IF-THEN. As saídas das regras são então combinadas em uma função de pertinência que reflete o *grau de desejo*  $D$  do esquema de controle implementado. A função de pertinência resultante é *defuzzyficada* para calcular uma única saída de controle. Vários esquemas de controle podem ser compostos utilizando-se operadores da lógica *fuzzy*.

Embora o processo de *defuzzyficação* possa produzir resultados satisfatórios na maioria dos casos, em outros casos no entanto, onde existe conflito entre dois esquemas de controle, pode ser que produza resultados não apropriados para ambos. Para contornar este problema, em [7] se propõe associar a cada Esquema de Controle um contexto, que define as condições sob as quais cada controlador é aplicável. As condições contextuais são implementadas por um conjunto de regras que ativam os Esquemas de Controle, do tipo

$$IF \text{ contexto} = A \text{ THEN } \text{ative\_Esquema\_de\_Controle } C$$

onde  $A$  é um termo nebuloso representando o contexto real. Esta regra *fuzzy* ativa o esquema de controle  $C$  a um nível determinado pelo grau de verdade do antecedente da regra. O nível de ativação é utilizado para ponderar o *grau de desejo* do Esquema de Controle.

Os Esquemas Comportamento são estruturas que conectam certos padrões de ações, implementados por esquemas de controle, com certos estímulos provenientes do ambiente. Dessa forma, Comportamentos estímulo-resposta podem ser formulados para atingirem objetivos específicos. Um Comportamento é descrito como uma tripla  $B = (C,D,O)$ , onde  $D$  é um esquema de controle para um tipo específico de movimento,  $O$  é um conjunto de descritores de um objeto com respeito ao qual o movimento deve ser executado, e  $C$  é uma condição de contexto, que define o contexto de aplicabilidade do movimento. Descritores do objeto são modelos de objetos no mundo real e eles fazem com que o Comportamento atue com respeito ao mundo externo, em oposição a um esquema de controle, que atua de acordo com variáveis internas. Os esquemas Comportamento também podem ser compostos em esquemas Comportamentos mais complexos utilizando a mesma técnica de composição utilizada pelos esquemas de controle. Finalmente, os Planificadores (baseados na teoria clássica de inteligência artificial) podem utilizar especificações dos Comportamentos, em termos de pré-condições e pós-condições, para construir planos que permitam atingir objetivos específicos.

Em resumo, trata-se de uma arquitetura síncrona, com uma codificação de resposta discreta e um método de coordenação entre Comportamentos denominado *context-dependent blending*.

### 1.2.5 **Multiple Objective Behavior-Based Control [38][39]**

Este ASM baseia-se na teoria *Multiple Objective Decision Making*, que provê abordagens para a tomada de decisões em situações onde mais de um objetivo deve ser considerado. As técnicas tradicionais de otimização não são adequadas para este tipo de situação [39], porque quando os diversos objetivos envolvidos são conflitantes ou competitivos, elas favorecem um deles em detrimento dos outros. Matematicamente, um problema de múltiplos objetivos pode ser representado como

$$\hat{\mathbf{x}} = \max_{\mathbf{x} \in \mathbf{X}} [o_1(\mathbf{x}), o_2(\mathbf{x}), \dots, o_n(\mathbf{x})]$$

onde,  $\mathbf{x} = (x_1, x_2, \dots, x_N) \in \mathcal{R}^N$  é um vetor de variáveis de decisão, ou alternativas,  $N$ -dimensional. Os objetivos do sistema são representados por  $n$  funções “*objetivo*”,  $o_1, o_2, \dots, o_n$ . O grau de atendimento do  $k$ -ésimo objetivo pela alternativa  $\mathbf{x}$  é dado por  $o_k(\mathbf{x})$ , onde  $\mathbf{X} \subseteq \mathcal{R}^N$  define o conjunto de “alternativas possíveis”. Na literatura, este problema é conhecido como *problema de otimização vetorial*.

Uma vez que soluções ótimas podem não existir, deve-se buscar soluções para as quais cada objetivo seja satisfeito a um determinado grau. Desta maneira, uma solução *satisfatória* de Simon [22] estendida para o problema de otimização vetorial, consiste em um subconjunto reduzido do conjunto de alternativas possíveis, as quais excedam todos os níveis de aspiração (graus de satisfação) de cada um dos objetivos [39].

Outra noção de otimização muito útil no contexto da teoria *Multiple Objective Decision Making* consiste na Pareto-Otimização. Define-se, portanto, que  $\mathbf{x}^*$  é uma solução Pareto-Ótima para um problema de otimização vetorial se não existe nenhum  $\mathbf{x} \in \mathbf{X}$  tal que  $o_i(\mathbf{x}^*) \leq o_i(\mathbf{x})$  para todo  $i$  e  $o_j(\mathbf{x}^*) < o_j(\mathbf{x})$  para ao menos um  $j$  [39].

Intuitivamente, uma solução Pareto-Ótima é tal que nenhuma outra solução possa melhorar o grau de satisfação de um objetivo sem que simultaneamente deteriore ao menos um dos demais objetivos. Uma solução Pareto-Ótima também é conhecida como uma solução não-dominada, uma solução não-inferior, ou uma solução *eficiente* [39]. Dessa maneira, baseado em um julgamento subjetivo dos valores das funções “*objetivo*”, é possível excluir o conjunto de alternativas que não representam soluções Pareto-Ótimas. Além de excluir soluções ineficientes, este processo reduz significativamente o número de alternativas, o que simplifica a busca subsequente por soluções *satisfatórias*.

Neste ASM, um problema de controle complexo é dividido em um conjunto de problemas de controle mais simples de resolver. Isto corresponde a dividir tarefas complexas em objetivos específicos, o que permite relacionar Comportamentos com objetivos. O que se busca é encontrar soluções *satisfatórias* para problemas que envolvem a tomada de decisões quando múltiplos objetivos devem ser considerados.

O ASM *Multiple Objective Behaviour-Based Control* é implementado através dos seguintes passos:

- Determinação do repertório de Comportamentos. Dadas as especificações do sistema e os requisitos da tarefa, formular um conjunto de objetivos específicos  $o_1, o_2, \dots, o_n$  e associar atributos mensuráveis que indiquem o grau de atendimento de cada um dos objetivos. Cada objetivo é implementado por um Comportamento, e, portanto, os objetivos definem o repertório de Comportamentos necessário para um determinado sistema.
- Projeto dos Comportamentos. Cada função “*objetivo*”  $o_i$  é modelada como um Comportamento  $b_i$  que mede os valores dos atributos para aquele objetivo. Assim, o Comportamento  $b_i$  é responsável pelo  $i$ -ésimo objetivo  $o_i$ .
- Seleção da ação. A partir do conjunto de ações ou alternativas possíveis,  $\mathbf{X}$ , seleciona-se as ações que consistam em soluções Pareto-Ótimas e *satisfatórias* para o problema de otimização vetorial em questão, onde as funções “*objetivo*”  $o_1, o_2, \dots, o_n$ , são as que foram identificadas no primeiro passo e foram implementadas pelos Comportamentos  $b_1, b_2, \dots, b_n$ , no segundo passo.

Em resumo, nesta arquitetura de controle, cada Comportamento calcula uma função “*objetivo*” para o conjunto de ações de controle admissíveis. A ação que maximiza a função “*objetivo*” corresponde à ação que melhor satisfaz o objetivo do Comportamento. Os múltiplos Comportamentos são mesclados em um único Comportamento mais complexo que busca selecionar a ação que simultaneamente satisfaz aos objetivos de cada Comportamento da melhor maneira possível, o que consiste em um método de otimização vetorial. Trata-se, portanto, de uma arquitetura síncrona que apresenta uma codificação de resposta discreta e um método de coordenação entre os Comportamentos baseado na teoria conhecida como *Multiple Objective Decision Making* [55].

### 1.2.6 Controle Concorrente [9]

Os criadores deste ASM consideram que a partir de vários controladores ruidosos, e sem realizar arbitração de comandos, é possível que um robô móvel apresente um comportamento estável e previsível pelo simples fato de que os motores do robô tendem a responder à média temporal das suas entradas. Desse modo, vários agentes (controladores), possivelmente dispersos em rede, podem controlar concorrentemente um único robô sem a necessidade de comunicação explícita. Tais agentes podem ser diferentes sob vários aspectos, desde o tipo de dados que utilizam, até a lei de controle empregada.

Este ASM pode ser classificado como um método de fusão de comandos baseado em superposição. No entanto, ele difere dos outros métodos baseados em superposição pelo fato de que a superposição é realizada fisicamente pelos motores, ao invés de ser realizada por um microprocessador.

Para que esta arquitetura de controle seja viável, é necessário que as seguintes condições sejam satisfeitas:

1. A planta deve ter uma inércia “significativa”;
2. Os sinais de controle devem ser produzidos a uma alta frequência;
3. Os sinais de controle a serem combinados devem ser adequados à tarefa em questão;
4. Deve existir uma polarização nos sinais de entrada que “aponte para a direção correta”.

A condição 1 é satisfeita por muitos sistemas físicos. Porém, poucos sistemas simulados satisfazem esta condição. A condição 2 quase sempre pode ser satisfeita, a menos que existam restrições de capacidade de processamento ou de comunicação. A condição 3 é a mais restritiva, pois restringe o elenco das tarefas executáveis àquelas em que o comportamento desejável do sistema consiste em uma forma de *servo*, ou seja, quando o sistema busca controlar uma determinada variável, como posição ou velocidade, por exemplo. Esta limitação ainda permite a execução de tarefas comuns, tais como seguimento de trajetória e busca de um ponto de destino. A princípio, qualquer tarefa de alto nível pode ser decomposta em componentes tipo *servo*. Tal decomposição, porém, pode vir a ser muito inconveniente. A condição final consiste de fato em uma exigência que deve ser satisfeita pelos controladores envolvidos. Uma vez que o sistema irá responder à média dos sinais de controle, a média dos sinais produzidos pelos controladores deve atingir o objetivo da tarefa. Desse modo, no clássico exemplo em que um obstáculo é detectado à frente do robô, enquanto a metade dos controladores envia sinais para desviar à esquerda, a outra metade envia sinais para desviar à direita. Assim sendo, esta arquitetura (assim como outras baseadas em campos potenciais) guiará o robô em linha reta, revelando-se, portanto, incapaz de evitar a colisão com o obstáculo detectado.

Em resumo, trata-se de uma arquitetura assíncrona que pode ser classificada como um método de fusão de comandos baseado em superposição, com uma codificação de resposta contínua, e que não utiliza nenhum método de coordenação entre Comportamentos, deixando isso a cargo da inércia dos motores.

### **1.3 Definição do Problema**

O problema que se deseja resolver nesta Tese de Doutorado é o da navegação de robôs móveis em ambientes semi-estruturados<sup>2</sup> utilizando para tanto uma arquitetura de controle reativa. Assume-se que entre o ponto de partida do robô e a posição final desejada existe ao menos um caminho livre que permita a passagem do robô. Deseja-se também que a arquitetura de controle a ser utilizada defina uma metodologia para o projeto do sistema de controle do robô e para este seja possível realizar algum tipo de análise de estabilidade.

---

<sup>2</sup> Define-se como ambiente semi-estruturado um ambiente interior que é plano, fechado, sem nenhum buraco, mas que admite a presença de objetos móveis e pessoas em seu interior e cuja configuração das paredes e objetos é desconhecida.

## **1.4 Objetivo**

O objetivo desta Tese de Doutorado é propor uma arquitetura de controle reativa para navegação de robôs móveis em ambientes desconhecidos e fechados, dentro do contexto de sistemas de controle não-lineares. Tal arquitetura se baseia na fusão de sinais de controle provenientes de diversos controladores, utilizando para tanto o Filtro de Informação Descentralizado.

### **1.4.1 Tarefas**

Para atingir o objetivo deste trabalho, as seguintes tarefas devem ser realizadas:

- 1) Fazer uma análise dos Filtros de Kalman [1][2][45] e de Informação [1], e a partir daí, definir o Filtro de Informação Descentralizado [13].
  - a) Estudo dos Filtros de Kalman.
  - b) Estudo dos Filtros de Informação.
  - c) Derivação do Filtro de Informação Descentralizado.
- 2) Desenvolvimento e análise de algoritmos de controle estáveis.
  - a) Estudo de alguns algoritmos de controle já disponíveis para navegação de robôs móveis, incluindo sua análise de estabilidade.
  - b) Desenvolvimento de um algoritmo para executar a tarefa de navegação em corredores e para seguimento de paredes, baseado em informações de distância às paredes obtidas com sensores de ultra-som, e demonstração de sua estabilidade.
  - c) Adaptação do algoritmo de controle de impedância aplicado a robôs móveis [24][25][48] com o fim de obter um algoritmo de controle para desempenhar a tarefa de evitar obstáculos, com demonstração de estabilidade no sentido de Lyapunov.
- 3) Descrever a técnica de navegação de robôs móveis baseada em fusão de sinais de controle provenientes de diversos controladores [13][14][16][17][18][19][20].
  - a) Escolha dos controladores a serem utilizados na fusão.
  - b) Proposição da técnica de fusão de sinais de controle.
  - c) Análise de estabilidade da fusão de sinais de controle provenientes de diversos controladores.



- 4) Obter conclusões sobre a estabilidade do sistema de controle resultante da fusão das saídas de diversos controladores.
  - a) Estudar o caso onde os controladores envolvidos no processo de fusão têm o mesmo objetivo de controle.
  - b) Estudar o caso onde os controladores envolvidos no processo de fusão têm objetivos de controle diferentes.
- 5) Análise do desempenho dos controladores e da fusão de sinais de controle.
  - a) Simulações.
  - b) Experimentos práticos utilizando o robô móvel PIONEER 2DX.

## **1.5 Contribuições desta Tese de Doutorado**

As principais contribuições desta Tese de Doutorado são:

- Derivação do Filtro de Informação Descentralizado, a partir das equações do Filtro de Kalman, do Filtro de Kalman Descentralizado e do Filtro de Informação.
- Desenvolvimento de um sistema de controle estável (no sentido de Lyapunov) para o seguimento de paredes e a navegação no centro de corredores, baseando-se em informações fornecidas por sensores de ultra-som.
- Desenvolvimento de um sistema de controle estável (no sentido de Lyapunov) para evitar obstáculos, baseado no controle de impedância aplicado a robôs móveis.
- Desenvolvimento de uma nova arquitetura de controle para navegação de robôs móveis, baseada na fusão das saídas de diferentes controladores, para a qual foi realizada uma análise detalhada de estabilidade.

Parte desta Tese de Doutorado, sobretudo os desenvolvimentos na área de controle não-linear e os experimentos realizados, foi realizada no Instituto de Automática – INAUT, da Universidad Nacional de San Juan – UNSJ na cidade de San Juan, Argentina, através de um convênio de cooperação existente entre a Universidade Federal do Espírito Santo (Brasil) e Universidad Nacional de San Juan (Argentina), apoiado pelas instituições CAPES, do Brasil, e SETCIP, da Argentina.

## **1.6 Estrutura da Tese**

A presente Tese de Doutorado está assim dividida:

- **Capítulo 1: Introdução**

Neste capítulo introdutório, inicialmente são apresentados os trabalhos correlatos mais significativos. Em seguida são apresentados os objetivos desta Tese, a definição do problema a ser tratado, as principais contribuições desta Tese e a sua estrutura.

- **Capítulo 2: O Filtro de Informação Descentralizado**

Trata dos Filtros de Kalman e dos Filtros de Informação, mostrando como é derivado o Filtro de Informação Descentralizado.

- **Capítulo 3: A Arquitetura de Controle Proposta**

Neste capítulo é apresentada a nova arquitetura de controle para navegação de robôs móveis, objeto central desta Tese de Doutorado. Também é apresentada a interpretação que é dada às covariâncias associadas a cada controlador e como é utilizada a lógica *fuzzy* para o cálculo das covariâncias associadas a cada controlador.

- **Capítulo 4: Análise de Estabilidade**

Neste capítulo é realizada uma análise de estabilidade da arquitetura de controle proposta. Inicialmente é analisado o caso onde os controladores envolvidos no processo de fusão possuem o mesmo objetivo de controle. Para este caso foi possível a realização de uma análise de estabilidade rigorosa, no sentido de Lyapunov. O outro caso analisado é quando os controladores envolvidos no processo de fusão não possuem o mesmo objetivo de controle. Neste caso, é proposta uma conjectura para o “bom comportamento” do sistema, baseada em funções energia.

- **Capítulo 5: Controladores Utilizados**

Neste capítulo são apresentados os quatro controladores utilizados nesta Tese de Doutorado, e suas respectivas funções energia.

- **Capítulo 6: Simulações e Resultados Experimentais**

Este capítulo inicia com uma descrição do robô móvel PIONEER 2DX, utilizado como plataforma de testes. São apresentados simulações e experimentos que ilustram e comprovam o bom desempenho da arquitetura de controle proposta. Os experimentos apresentados neste capítulo também têm como objetivo comprovar a conjectura de “bom comportamento” do sistema quando este envolve a fusão de controladores que não possuem o mesmo objetivo de controle. No final do capítulo são discutidos os resultados das simulações e experimentos.

- **Capítulo 7: Conclusões**

Neste capítulo são apresentadas as conclusões desta Tese. Também é realizada a análise dos resultados obtidos, e são apresentadas algumas propostas para trabalhos futuros, seguindo esta linha de pesquisa.

- **Bibliografia**

Finalmente, é apresentada uma lista das Referências Bibliográficas mencionadas ao longo do texto e uma lista contendo a bibliografia consultada ao longo deste trabalho.

## Capítulo 2

# O Filtro de Informação Descentralizado

Neste capítulo é apresentada a formulação matemática dos Filtros de Kalman [2][45], em diversas formas, assim como dos Filtros de Informação [1], que são derivados dos Filtros de Kalman. A razão para discutir tais filtros neste capítulo é que os mesmos são ferramentas importantes para realizar a fusão de dados, que interessa a este trabalho.

O Filtro de Kalman permite a fusão de dados de dois ou mais sensores de uma maneira otimizada<sup>3</sup>, conforme ilustra a Figura 2.

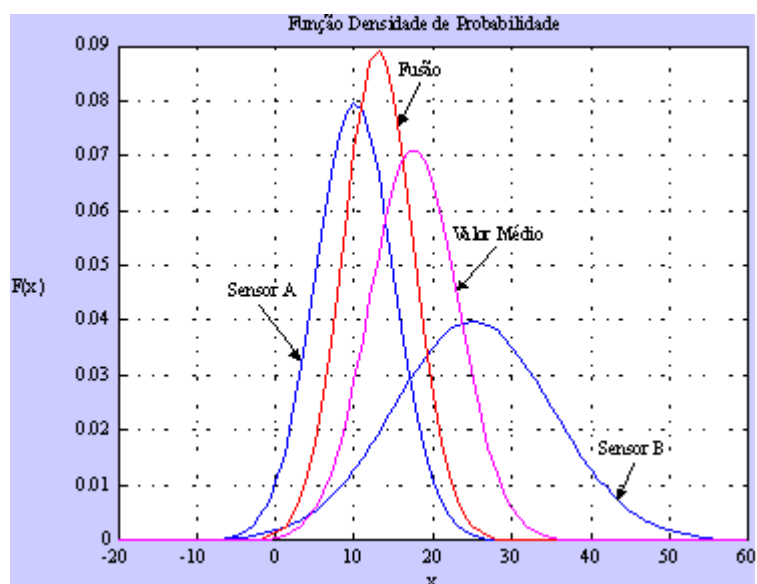


Figura 2 Comparação do resultado da fusão dos dados provenientes de dois sensores, usando Filtro de Kalman, com a simples média aritmética instantânea dos mesmos.

<sup>3</sup> Otimizada no sentido de que a variância dos dados de saída do filtro de Kalman é menor do que a menor das variâncias dos dados medidos por cada sensor individualmente.

Conforme mostrado na Figura 2, a variância da média aritmética instantânea dos dados medidos pelos dois sensores não é tão pequena quanto a variância do sensor mais preciso (sensor A), mas também não é tão grande quanto a variância do sensor menos preciso (sensor B). Por outro lado, a fusão dos sinais pela aplicação do Filtro de Kalman permite obter como resultado um conjunto de dados cuja variância é menor do que a variância de ambos os sensores, ou seja, os dados resultantes da fusão são mais confiáveis que os dados de cada sensor em separado. Este exemplo simples mostra a viabilidade de se usar o Filtro de Kalman para realizar fusão de dados.

No caso particular deste trabalho, será usado não o Filtro de Kalman, mas sim o Filtro de Informação [1], na sua versão descentralizada, por suas vantagens inerentes, que serão abordadas na sequência. A opção pela versão descentralizada é feita devido ao fato de que dessa maneira é possível evitar o uso de operações com matrizes.

Um filtro descentralizado pode ter duas abordagens: na primeira, o filtro deve ser implementado utilizando mais de uma unidade de processamento [1], como por exemplo no caso do problema da localização simultânea de um grupo de robôs capazes de detectarem a presença uns dos outros. Cada um dos robôs coleta dados sensoriais sobre a sua própria movimentação e compartilha esta informação com o resto do grupo. Neste caso pode-se utilizar um único estimador, na forma de um filtro descentralizado. Na segunda abordagem, o filtro descentralizado é composto por um filtro global e vários filtros locais que não necessitam ser implementados em unidades de processamento distintas. O Filtro de Kalman e o Filtro de Kalman Descentralizados são apresentados em [45], sendo que este último segue a segunda abordagem mencionada. Por outro lado, em [1], o Filtro de Informação e uma implementação do Filtro de Informação Descentralizado são apresentados. Neste caso, o Filtro de Informação Descentralizado segue a primeira abordagem previamente mencionada. Uma das contribuições originais desta Tese de Doutorado é a definição de um Filtro de Informação Descentralizado baseado na segunda abordagem descrita anteriormente. Para derivar este Filtro de Informação Descentralizado, inicialmente se apresentará o Filtro de Kalman, e em seguida o Filtro de Kalman Descentralizado. Posteriormente, será apresentado o Filtro de Informação. Finalmente, baseando-se nestes resultados, será derivado o Filtro de Informação Descentralizado utilizado neste trabalho.

## 2.1 O Filtro de Kalman

Seja um sistema modelado por

$$\mathbf{x}(k+1) = \boldsymbol{\Phi}(k)\mathbf{x}(k) + \mathbf{w}(k) \quad (1)$$

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \quad (2)$$

onde  $\mathbf{x}(k)$  é o vetor de estados  $n \times 1$  do processo no instante  $t_k$ ,  $\boldsymbol{\Phi}(k)$  é uma matriz  $n \times n$  relacionando  $\mathbf{x}(k)$  e  $\mathbf{x}(k+1)$ ,  $\mathbf{z}(k)$  é o vetor de observação  $m \times 1$  no instante  $t_k$  e  $\mathbf{H}(k)$  é a matriz de observação  $m \times n$  que descreve a conexão ideal (sem ruído) entre as medições e o vetor de estados no instante  $t_k$ . O vetor  $n \times 1$   $\mathbf{w}(k)$  consiste em uma sequência branca de ruídos gaussianos com covariância conhecida, a qual representa o ruído associado ao sistema, e  $\mathbf{v}(k)$  é um vetor  $m \times 1$  representando o erro de medição, também consistindo em uma sequência

branca de ruídos gaussianos, com covariância conhecida. A correlação cruzada entre os vetores  $\mathbf{v}(k)$  e  $\mathbf{w}(k)$  é zero, e as matrizes de covariância associadas a eles são dadas por

$$E[\mathbf{w}(i)\mathbf{w}^T(k)] = \begin{cases} \mathbf{Q}(k) & i = k \\ 0 & i \neq k \end{cases} \quad (3)$$

$$E[\mathbf{v}(i)\mathbf{v}^T(k)] = \begin{cases} \mathbf{R}(k) & i = k \\ 0 & i \neq k \end{cases} \quad (4)$$

Considerando uma estimativa inicial do processo  $\hat{\mathbf{x}}(k|k-1)$  no instante  $t_k$  (a qual é baseada no conhecimento prévio sobre o processo antes de  $t_k$ ), as equações correspondentes ao Filtro de Kalman são

Predição:

$$\hat{\mathbf{x}}(k|k-1) = \boldsymbol{\Phi}(k)\hat{\mathbf{x}}(k-1) \quad (5)$$

onde  $\hat{\mathbf{x}}(k|k-1)$  é a previsão de  $\mathbf{x}(k)$ , dadas as observações realizadas até o instante  $k-1$ .

$$\mathbf{P}(k|k-1) = \boldsymbol{\Phi}(k)\mathbf{P}(k-1)\boldsymbol{\Phi}^T(k) + \mathbf{Q}(k) \quad (6)$$

onde  $\mathbf{P}(k|k-1)$  é a previsão de  $\mathbf{P}(k)$ , dadas as observações realizadas até o instante  $k-1$ .

Estimação:

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1)) \quad (7)$$

onde  $\hat{\mathbf{x}}(k)$  é a previsão de  $\mathbf{x}(k)$ , dadas as observações realizadas até o instante  $k$ .

$$\mathbf{P}^{-1}(k) = (\mathbf{P}(k|k-1))^{-1} + \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k) \quad (8)$$

onde  $\mathbf{P}(k)$  é a previsão de  $\mathbf{P}(k)$ , dadas as observações realizadas até o instante  $k$ .

$$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{H}^T(k)\mathbf{R}^{-1}(k)$$

onde  $\mathbf{P}(k)$  é a matriz de covariância do erro entre o estado real  $\mathbf{x}(k)$  e o estado estimado  $\hat{\mathbf{x}}(k)$ . Este filtro é mostrado na Figura 3, na forma de diagrama de blocos e  $\mathbf{K}$  é o ganho de Kalman.

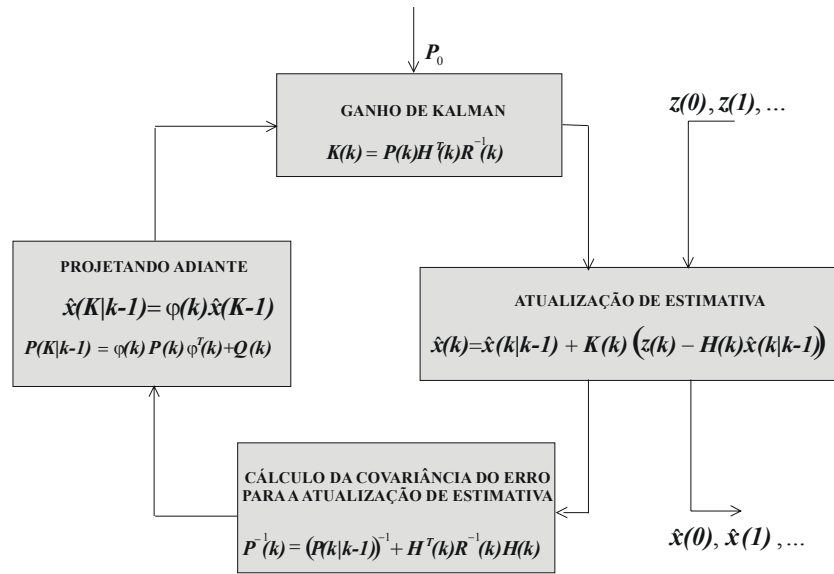


Figura 3 O Filtro de Kalman [45].

## 2.2 O Filtro de Kalman Descentralizado

Eventualmente, existem aplicações onde um filtro centralizado não pode ser obtido [45]. Uma alternativa nestes casos é a utilização de filtros descentralizados. Um filtro descentralizado pode trazer ainda a vantagem de evitar o uso de matrizes, o que simplifica os cálculos.

Para o  $i$ -ésimo filtro local tem-se

$$\hat{\mathbf{x}}_i(k) = \mathbf{P}_i(k) \left( \mathbf{P}^{-1}(k-1) \hat{\mathbf{x}}(k-1) + \mathbf{H}_i(k) \mathbf{R}_i^{-1}(k) \mathbf{z}_i(k) \right) \quad (9)$$

$$\mathbf{P}_i^{-1}(k) = \mathbf{P}^{-1}(k-1) + \mathbf{H}_i^T(k) \mathbf{R}_i^{-1}(k) \mathbf{H}_i(k) \quad (10)$$

onde  $\mathbf{P}(k-1)$  é a matriz de covariância do erro no instante  $k-1$  e  $\hat{\mathbf{x}}(k-1)$  é o vetor de estados estimados. Para o filtro global, no instante  $k$ , tem-se

$$\hat{\mathbf{x}}(k) = \mathbf{P}(k) \left[ \sum_{i=1}^n \mathbf{P}_i^{-1}(k) \hat{\mathbf{x}}_i(k) - (n-1) \mathbf{P}^{-1}(k-1) \hat{\mathbf{x}}(k-1) \right] \quad (11)$$

$$\mathbf{P}^{-1}(k) = \sum_{i=1}^n \mathbf{P}_i^{-1}(k) - (n-1) \mathbf{P}^{-1}(k-1) \quad (12)$$

onde  $n$  é o número de filtros locais. A Figura 4 ilustra o esquema do Filtro de Kalman Descentralizado, com dois filtros locais.

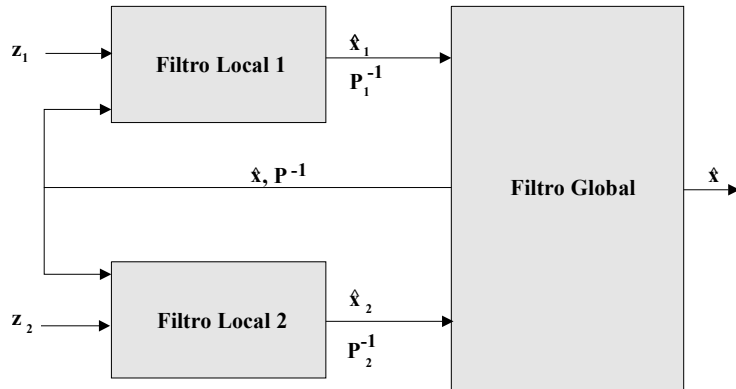


Figura 4 Exemplo de Filtro de Kalman Descentralizado com dois filtros locais.

Também existe uma versão do Filtro de Kalman Descentralizado onde não existe o laço de realimentação de  $\hat{\mathbf{x}}$  e  $\mathbf{P}^{-1}$  (laço este mostrado na Figura 4). Não se pode, porém, garantir que ele produz exatamente os mesmos resultados do que o Filtro de Kalman centralizado (mesmo da seção 2.1), já que cada filtro local não tem acesso a nenhuma informação processada pelos demais. Por esta razão o Filtro de Kalman Descentralizado sem realimentação não foi aqui apresentado (mais detalhes sobre ele podem ser obtidos em [45]).

## 2.3 O Filtro de Informação [1]

O Filtro de Informação é essencialmente um Filtro de Kalman expresso em termos de medidas de informação sobre os estados de interesse ao invés de estimativas dos estados e suas covariâncias associadas [1]. Este filtro também é conhecido como a *forma de covariância inversa* do filtro de Kalman [1]. As vantagens do filtro de informação com relação ao filtro de Kalman são apresentadas na Seção 2.5.

As equações do filtro de Informação são assim definidas

Matriz de Informação:

$$\mathbf{Y}(k) = \mathbf{P}^{-1}(k) \quad (13)$$

Vetor de Informação de Estado:

$$\hat{\mathbf{y}}(k) = \mathbf{P}^{-1}(k)\hat{\mathbf{x}}(k) = \mathbf{Y}(k)\hat{\mathbf{x}}(k) \quad (14)$$

As equações de predição e estimação do Filtro de Informação são

Predição:

$$\hat{\mathbf{y}}(k) = \mathbf{L}(k)\hat{\mathbf{y}}(k-1) \quad (15)$$

$$\mathbf{Y}(k) = [\boldsymbol{\Phi}(k)\mathbf{Y}^{-1}(k-1)\boldsymbol{\Phi}^T(k) + \mathbf{Q}(k)]^{-1} \quad (16)$$



Estimação:

$$\hat{\mathbf{y}}(k) = \hat{\mathbf{y}}(k|k-1) + \mathbf{i}(k) \quad (17)$$

$$\mathbf{Y}(k) = \mathbf{Y}(k|k-1) + \mathbf{I}(k) \quad (18)$$

onde

$$\mathbf{L}(k) = \mathbf{Y}(k)\boldsymbol{\Phi}(k)\mathbf{Y}^{-1}(k-1)$$

$$\mathbf{i}(k) = \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{z}(k) \quad (19)$$

$$\mathbf{I}(k) = \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k) \quad (20)$$

são o coeficiente de propagação da informação, a contribuição da informação do estado e a matriz de informação associada a cada estado, respectivamente.

## 2.4 O Filtro de Informação Descentralizado

Para derivar as equações do Filtro de Informação Descentralizado, deve-se começar a partir das equações apresentadas nas seções anteriores deste capítulo, e proceder da seguinte maneira:

1. Para cada filtro local, deve-se pré-multiplicar ambos os membros de (9) por  $\mathbf{P}_i^{-1}(k)$  e então incluir os resultados de (13), (14) e (19). O resultado é

$$\hat{\mathbf{y}}_i(k) = \hat{\mathbf{y}}_i(k-1) + \mathbf{i}_i(k) \quad (21)$$

2. Na sequência, partindo de (10) e utilizando as equações (13) e (20), pode-se obter

$$\mathbf{Y}_i(k) = \mathbf{Y}_i(k-1) + \mathbf{I}_i(k) \quad (22)$$

3. Para o filtro global, deve-se seguir um procedimento similar, obtendo as seguintes equações para o instante  $t_k$

$$\hat{\mathbf{y}}(k) = \sum_{i=1}^n \hat{\mathbf{y}}_i(k) - (n-1)\hat{\mathbf{y}}(k-1) \quad (23)$$

$$\mathbf{Y}(k) = \sum_{i=1}^n \mathbf{Y}_i(k) - (n-1)\mathbf{Y}(k-1) \quad (24)$$

onde  $n$  é o número de filtros locais.

O diagrama de blocos correspondente ao Filtro de Informação Descentralizado é idêntico ao apresentado na Figura 4 para o Filtro de Kalman Descentralizado.

## **2.5 Comparação entre o Filtro de Informação e o Filtro de Kalman**

Algumas das características importantes associadas ao Filtro de Kalman e ao Filtro de Informação e suas versões descentralizadas são comparadas nesta seção. A primeira comparação é quanto à forma de inicialização: apesar de ser equivalente ao Filtro de Kalman, o Filtro de Informação é mais fácil de inicializar. Quando se está utilizando um Filtro de Kalman, é necessário um conhecimento prévio a respeito do sistema com o objetivo de ter uma boa estimativa inicial para o vetor de estados e para a matriz de covariância do erro. Por sua vez, o Filtro de Informação é inicializado simplesmente considerando o vetor de informação de estado  $y$  igual a zero e a matriz de informação  $Y$  igual a um valor muito pequeno, porém diferente de zero, permitindo que ela possa ser invertida. Isto significa que nenhuma informação está disponível antes de iniciar a execução do filtro. De fato, a informação disponível aumenta ao longo do tempo. A vantagem é mais evidente quando se trata de sistemas não lineares, porque a convergência do Filtro de Kalman não é garantida quando não se dispõe de condições iniciais adequadas. Em outras palavras, no que diz respeito a inicialização, o Filtro de Informação é mais robusto do que o Filtro de Kalman.

Outra característica é a complexidade do filtro. Enquanto as equações de predição do Filtro de Informação (equações (15) e (16)) são mais complexas do que as equações equivalentes do Filtro de Kalman (equações (5) e (6)), as equações de estimação daquele (equações (17) e (18)) são muito mais simples do que as equações correspondentes deste (equações (7) e (8)). Assim, a formulação do Filtro de Informação Descentralizado é mais simples do que a formulação do Filtro de Kalman Descentralizado, o que pode ser verificado comparando as equações (21)-(24) com as equações (9)-(12), que dizem respeito ao Filtro de Informação Descentralizado e ao Filtro de Kalman Descentralizado, respectivamente.

Além disso, no caso do Filtro de Informação, a matriz de mais alta ordem a ser invertida tem a mesma dimensão do vetor de estados. Desse modo, para sistemas multi-sensoriais a dimensão máxima das matrizes a serem invertidas seria bem menor do que a dimensão do vetor de observação, que define a dimensão máxima das matrizes a serem invertidas quando se utiliza o Filtro de Kalman.

Também é necessário mencionar que o uso de Filtros de Informação torna mais simples a criação de sistemas distribuídos e descentralizados. Isto se deve ao fato de que é mais simples trocar informações sobre estados do que trocar informações entre nós destes sistemas [1]. O uso de Filtros de Informação Descentralizados permite a criação de sistemas distribuídos e descentralizados que não necessitam estar completamente conectados. Isto significa que somente os nós que efetivamente necessitam trocar informações entre si necessitam estar conectados [1]. Além disso, é possível garantir que as mensagens a serem trocadas serão as menores possíveis [1].

A vantagem de utilizar as versões descentralizadas do Filtro de Kalman e do Filtro de Informação consiste em não se necessitar inverter matrizes. Para isso é necessário utilizar uma quantidade de filtros locais igual ao número de fontes de informação cujos dados se deseja fusionar. Dessa maneira, as matrizes se reduzem a números escalares, e todo o processo é simplificado.

Devido às vantagens associadas ao Filtro de Informação Descentralizado (FID), como destacadas anteriormente, ele foi selecionado como o método de fusão utilizado na arquitetura de controle aqui proposta. São utilizados dois Filtros de Informação Descentralizados, um para realizar a fusão das velocidades angulares e outro para realizar a fusão das velocidades lineares. Na seção seguinte, é apresentado um estudo de caso com o objetivo de demonstrar a equivalência entre o Filtro de Kalman, o Filtro de Kalman Descentralizado, o Filtro de Informação e o Filtro de Informação Descentralizado.

## **2.6 Filtro de Kalman, Filtro de Kalman Descentralizado, Filtro de Informação e Filtro de Informação Descentralizado: Um Estudo de Caso**

Para realizar um estudo comparativo do desempenho do Filtro de Kalman, do Filtro de Kalman Descentralizado, do Filtro de Informação e do Filtro de Informação Descentralizado, foi realizado um experimento envolvendo quatro sensores de ultra-som. Cada sensor foi utilizado para realizar um conjunto de 1000 medidas de distância a uma parede orientada perpendicularmente ao seu eixo de emissão acústica. A medição dos dados utilizados para este experimento foram realizadas por C. Soria em [10]. A distância real dos sensores à parede é de 1500 mm. Para efeito de comparação com os dados produzidos pelos filtros, foi calculada a média instantânea das medidas dos quatro sensores.

O sistema em questão foi modelado utilizando as equações (1)-(4), onde o único estado é a distância à parede. No modelo em questão

$$\phi(k) = 1 \text{ e } w(k) = 0$$

Desse modo, (1) se reduz a

$$x(k+1) = x(k)$$

já que uma vez que os sensores estão a uma distância constante da parede, é de se esperar que o estado permaneça inalterado ao longo do tempo.

O vetor de observação  $z$  é composto por quatro observações (as leituras dos quatro sensores), e é dado por:

$$z(k) = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$$

onde  $d_1$ ,  $d_2$ ,  $d_3$  e  $d_4$ , são as medições de distância realizadas pelos quatro sensores de ultra-som.

Ainda no modelo em questão

$$\mathbf{H}(k) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \text{ e } \mathbf{R}(k) = \begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 21 & 0 \\ 0 & 0 & 0 & 260 \end{bmatrix}$$

Como mencionado anteriormente, os dados sensoriais necessários para este estudo de caso, o modelo do sistema e as matrizes  $\mathbf{H}(k)$  e  $\mathbf{R}(k)$  foram obtidos de [10].

Utilizando-se as equações apresentadas neste capítulo foram implementados em MATLAB<sup>®</sup> o Filtro de Kalman (FK), o Filtro de Kalman Descentralizado (FKD), o Filtro de Informação (FI), e o Filtro de Informação Descentralizado (FID).

A Tabela 1 apresenta a média e o desvio padrão dos dados produzidos por cada um dos quatro filtros implementados e também a média e o desvio padrão da média instantânea das medidas dos quatro sensores (MI).

Tabela 1 Média e desvio padrão dos dados medidos pelos quatro sensores, dos dados resultantes dos quatro filtros em questão e da média instantânea das medidas dos quatro sensores.

Fonte de Dados	Média (mm)	Desvio Padrão
Sensor 1	1483	2,3023
Sensor 2	1483	3,9849
Sensor 3	1486	4,5882
Sensor 4	1462	16,1469
MI	1479	4,3405
FK	1484	0,4504
FKD	1484	0,4504
FI	1484	0,4475
FID	1484	0,4475

A Figura 5 ilustra as funções de densidade de probabilidades dos dados resultantes da média instantânea dos quatro sensores, do Filtro de Kalman e do Filtro de Informação Descentralizado. As funções de distribuição de probabilidades dos dados produzidos pelo Filtro de Kalman Descentralizado e pelo Filtro de Informação foram omitidas, pois, conforme ilustra a Tabela 1, são idênticas às funções do Filtro de Kalman e do Filtro de Informação Descentralizado, respectivamente.

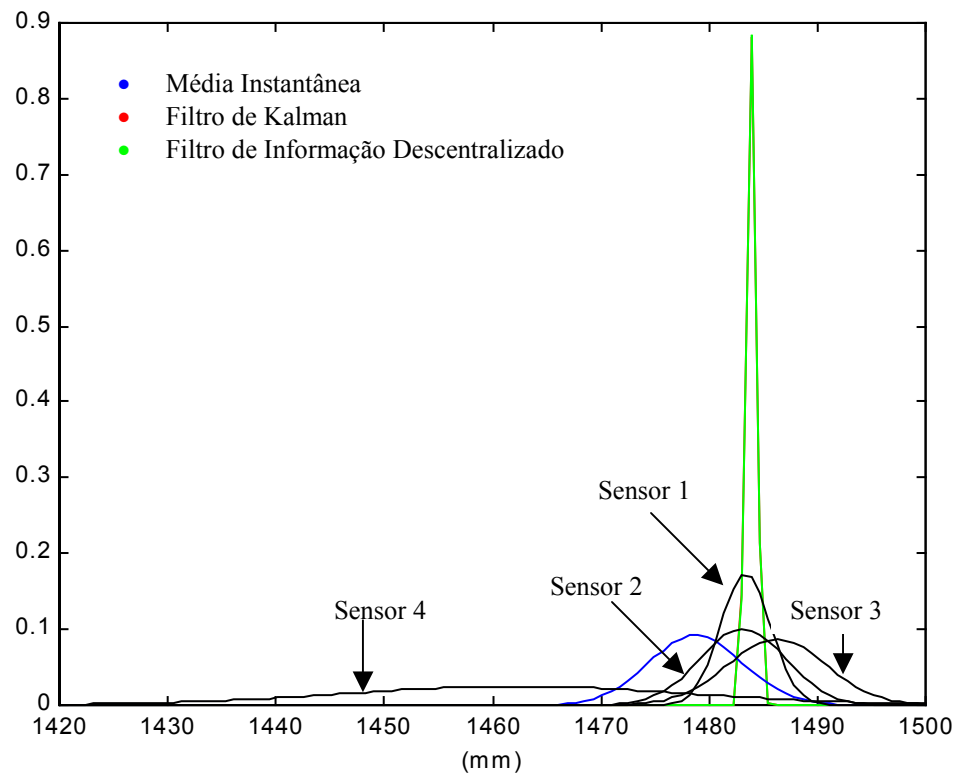


Figura 5 Funções de densidade de probabilidade da média instantânea das medidas dos sensores, do Filtro de Kalman, do Filtro de Informação Descentralizado<sup>4</sup> e dos quatro sensores de ultra-som.

Conforme ilustra a Figura 5, a função de densidade de probabilidade do Filtro de Kalman e do Filtro de Informação Descentralizado são quase idênticas, porém muito diferentes da função de densidade de probabilidade da média instantânea dos dados medidos pelos quatro sensores. Como se pode observar na Tabela 1 e na Figura 5, as médias dos dados obtidos pelo Filtro de Kalman e pelo Filtro de Informação Descentralizado são idênticas, e ligeiramente diferentes da média instantânea dos dados dos quatro sensores. No entanto, é possível verificar que o desvio padrão da média instantânea é muito maior do que os desvios padrões dos resultados obtidos pelo Filtro de Kalman e pelo Filtro de Informação Descentralizado. Na Tabela 1 é possível verificar que o desvio padrão dos dados obtidos pelo Filtro de Informação e pelo Filtro de Informação Descentralizado é um pouco menor do que o desvio padrão dos dados obtidos pelo Filtro de Kalman e pelo Filtro de Kalman Descentralizado.

A Figura 5 também mostra as funções de densidade de probabilidade dos quatro sensores utilizados no experimento, e fica claro que os resultados obtidos utilizando o Filtro de Kalman e o Filtro de Informação Descentralizado possuem um desvio padrão menor do que o menor dos desvios padrões dos quatro sensores, enquanto que o desvio padrão da média instantânea é menor que os desvios padrões dos dados medidos pelos sensores 3 e 4 e maior do que o desvio padrão dos dados medidos pelos sensores 1 e 2.

<sup>4</sup> Conforme é possível constatar pela Tabela 1, as funções de densidade de probabilidade do Filtro de Kalman e do Filtro de Informação são semelhantes e portanto não podem ser distinguidas entre si na Figura 5.

A seguir são apresentados os gráficos das medições dos quatro sensores e os resultados obtidos pelos quatro filtros em questão e pela média instantânea dos dados obtidos pelos quatro sensores. A Figura 6 ilustra os dados obtidos pelos quatro sensores de ultra-som e a Figura 7 mostra o resultado da média instantânea dos dados obtidos pelos sensores.

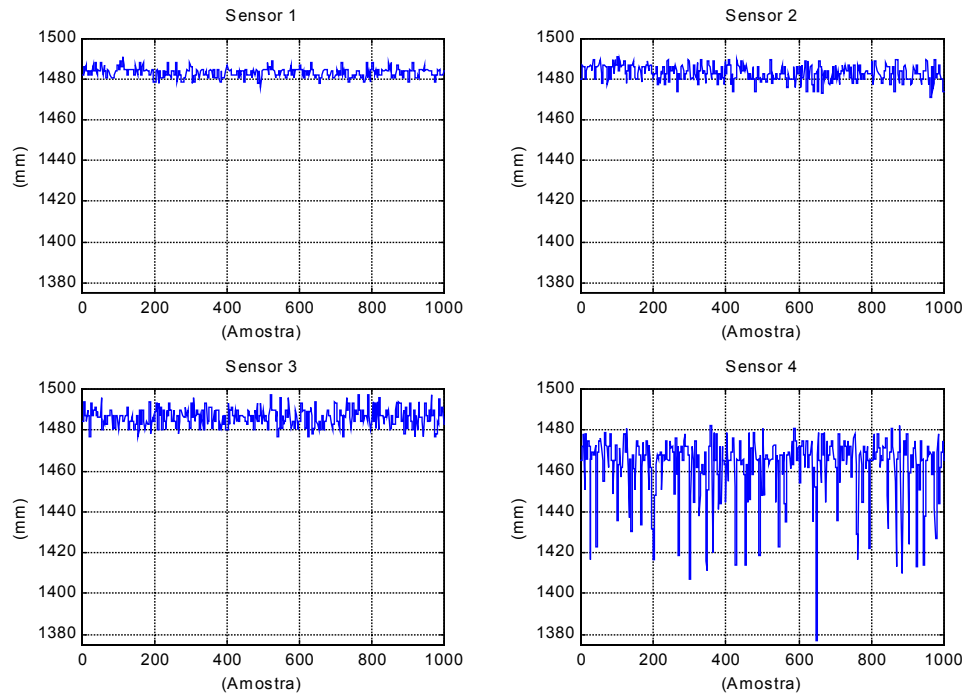


Figura 6 Dados medidos pelos quatro sensores.

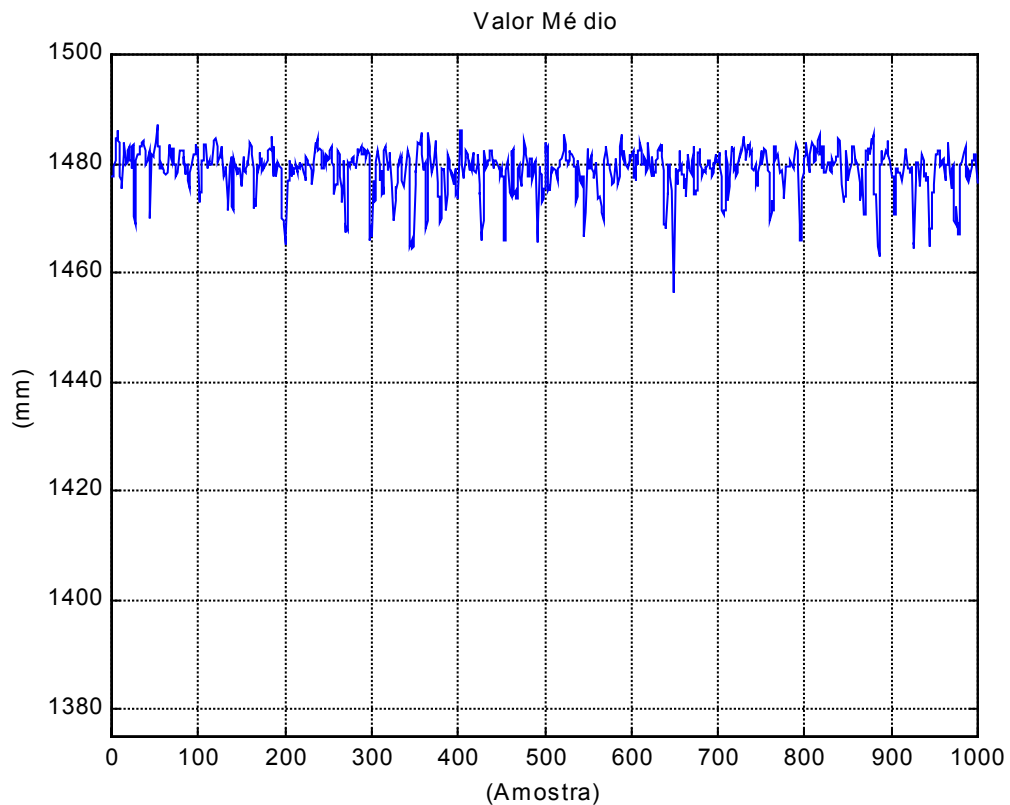


Figura 7 Média instantânea dos dados obtidos pelos quatro sensores.

A Figura 8 apresenta os resultados obtidos pela aplicação do Filtro de Kalman (FK), Filtro de Kalman Descentralizado (FKD), Filtro de Informação (FI) e Filtro de Informação Descentralizado (FID).

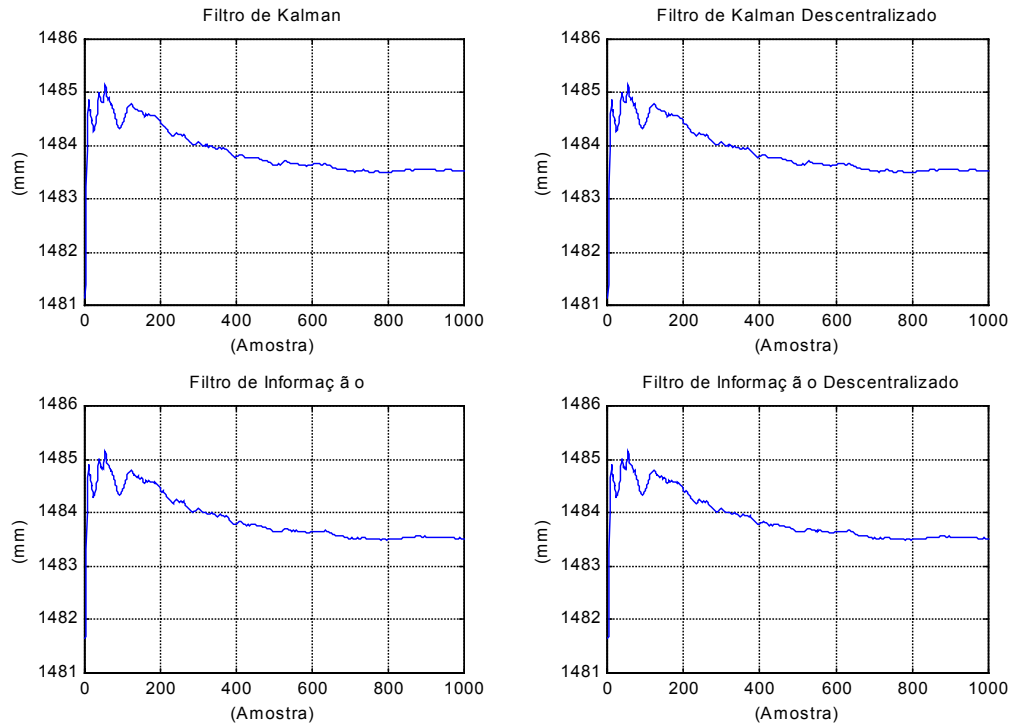


Figura 8 Resultados obtidos pelos filtros FK, FKD, FI e FID.

A Tabela 2 mostra a correlação cruzada, que indica o grau de similaridade dos resultados obtidos, para o Filtro de Kalman (FK), o Filtro de Kalman Descentralizado (FKD), o Filtro de Informação (FI) e o Filtro de Informação Descentralizado (FID).

Tabela 2 Correlações cruzadas entre os dados obtidos pelos filtros analisados quando tomados dois a dois.

Correlação Cruzada	FK	FKD	FI	FID
<b>FK</b>	1,0000	1,0000	0,9989	0,9989
<b>FKD</b>	1,0000	1,0000	0,9989	0,9989
<b>FI</b>	0,9989	0,9989	1,0000	1,0000
<b>FID</b>	0,9989	0,9989	1,0000	1,0000

Na Tabela 2 o fato da correlação cruzada entre os dados obtidos por dois filtros ser igual a 1 indica que os mesmos dados foram obtidos pelos dois filtros em questão. Quanto menor o valor da correlação cruzada, maior é a diferença entre os dados obtidos pelos dois filtros considerados.

Conforme ilustram as figuras anteriores, a Tabela 1 e a Tabela 2, os resultados obtidos pelo Filtro de Kalman e pelo Filtro de Kalman Descentralizado são idênticos. O mesmo acontece com os dados obtidos pelo Filtro de Informação e pelo Filtro de Informação Descentralizado. Porém, os dados obtidos pelo Filtro de Kalman e pelo Filtro de Informação são um pouco diferentes. Uma vez que Mutambara [1] afirma que os dois filtros são

algebricamente equivalentes, a pequena diferença verificada provavelmente é devida a erros de aproximação nos cálculos realizados. No entanto, qualquer um dos filtros em questão produz resultados bem melhores do que os resultados obtidos pela média instantânea dos dados obtidos pelos quatro sensores.

Com o objetivo de comparar a complexidade algébrica do o filtro de Kalman (FK), filtro de Kalman descentralizado (FKD), filtro de informação (FI) e filtro de informação descentralizado (FID), foi utilizada a função FLOPS do MATLAB<sup>®</sup>. Tal função conta o número de operações de ponto flutuante realizadas em um trecho de programa. O resultado é apresentado na Tabela 3.

Tabela 3 Número de FLOPS resultante quando se utilizam os filtros FK, FKD, FI e FID

Filtro	FLOPS
FK	360
FKD	97
FI	335
FID	55

Observando os resultados apresentados na Tabela 3, é possível chegar a três conclusões, (dentro do contexto do exemplo em questão):


- I. O filtro de informação de fato apresenta um menor número de FLOPS do que o filtro de Kalman, o que corrobora a afirmação feita por Mutambara [1] de que as equações do filtro de informação são mais simples do que as do filtro de Kalman.
- II. As versões descentralizadas dos filtros de Kalman e de informação apresentam um menor número de FLOPS do que as suas respectivas implementações centralizadas, o que indica que a utilização de escalares ao invés de matrizes, possível nos filtros descentralizados, permite uma redução de complexidade algébrica.
- III. Finalmente, verifica-se que o menor número de FLOPS ocorreu quando se utilizou o filtro de informação descentralizado proposto neste trabalho, o que indica que ele tem a menor complexidade algébrica, ao menos dentro do escopo do experimento em questão.

Assim, o Filtro de Informação Descentralizado é que foi escolhido como mecanismo de fusão neste trabalho, considerando que ele de fato realiza uma fusão otimizada dos dados, assim como as vantagens que ele apresenta sobre o Filtro de Kalman Descentralizado.



## Capítulo 3

# A Arquitetura de Controle Proposta



Neste capítulo é apresentada uma nova arquitetura de controle para a navegação de robôs móveis, que é o objeto central desta Tese de Doutorado. A arquitetura de controle proposta é adequada para qualquer aplicação onde vários objetivos de controle, muitas vezes conflitantes, necessitam ser considerados ao mesmo tempo. A partir deste ponto de vista, a aplicação da arquitetura de controle proposta neste trabalho para o controle da navegação de robôs móveis é um estudo de caso, que visa demonstrar o seu desempenho e as suas características.

A arquitetura proposta se inspira no fato de que, em muitos casos, os dados coletados pelos diversos sistemas de sensoriamento disponíveis em um robô móvel são tão distintos entre si<sup>5</sup>, que não só dificultam como até mesmo impossibilitam o uso de fusão sensorial. Nestes casos, é muito mais fácil projetar um controlador para cada tipo de dados de entrada e então realizar a fusão das saídas produzidas por tais controladores. Como as saídas produzidas por todos os controladores consistem em valores de velocidades lineares e angulares, a fusão pode ser realizada sem maiores problemas.

A arquitetura de controle proposta neste trabalho se classifica como um ASM (*Action Selection Mechanism*) que se baseia na fusão de comandos, sendo que o método de fusão utilizado é o Filtro de Informação Descentralizado (FID), apresentado no capítulo anterior, e o número de filtros locais é igual ao número de controladores envolvidos no processo de fusão. São utilizados dois Filtros de Informação Descentralizados: um para realizar a fusão das velocidades angulares e outro para realizar a fusão das velocidades lineares. O conjunto de controladores a ser utilizado depende da aplicação específica à qual se destina o sistema de controle. No caso em que algum tipo de análise de estabilidade seja desejável, todos os controladores utilizados devem conduzir a sistemas de controle estáveis no sentido de Lyapunov.

---

<sup>5</sup> Por exemplo, medidas de distância obtidas através de sensores de ultra-som e medidas de fluxo-ótico obtidas a partir de imagens captadas por uma câmera de vídeo.

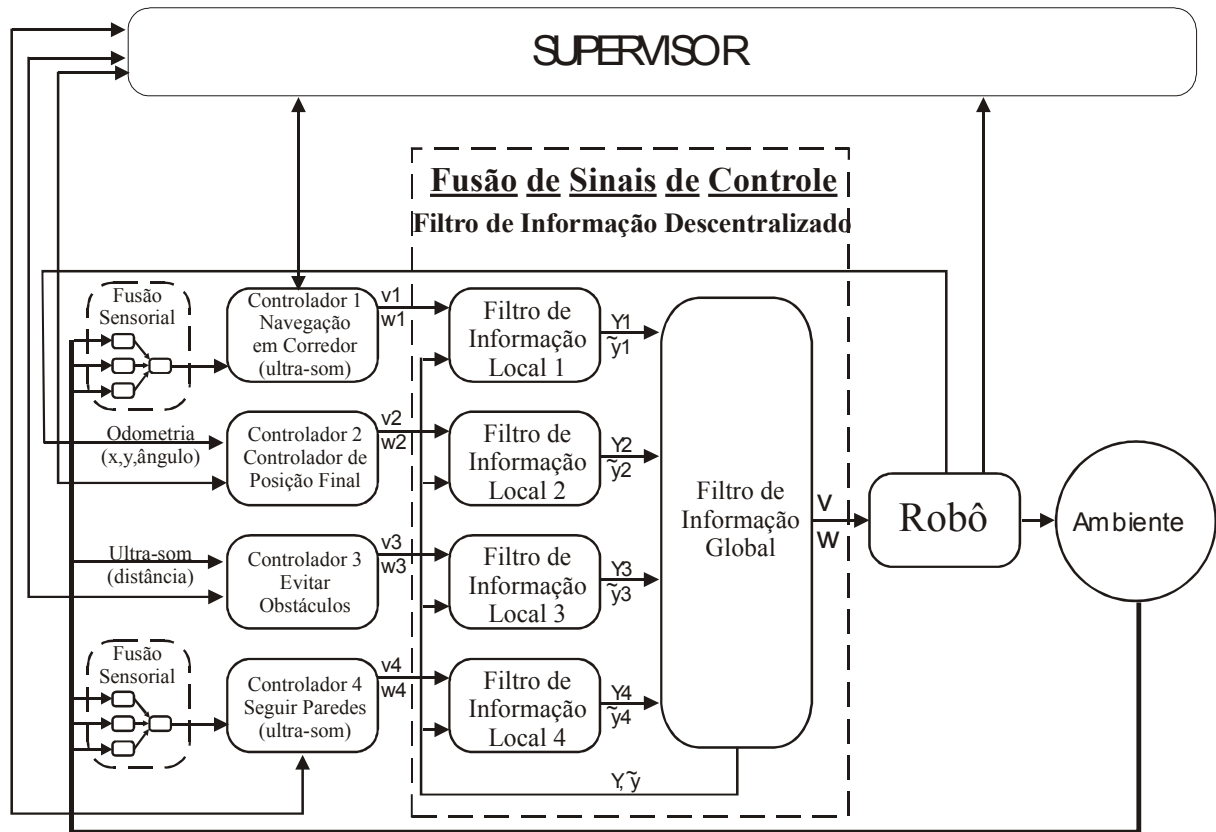


Figura 9 Implementação da arquitetura proposta para o controle da navegação de um robô móvel em um ambiente composto por corredores e obstáculos.

A arquitetura de controle proposta é síncrona, ou seja, as saídas dos controladores e o cálculo da saída do FID que é enviada ao robô ocorrem dentro do espaço de tempo de 100ms que é a taxa de atualização dos dados medidos pelos sensores. A princípio não haveria problemas se a arquitetura fosse assíncrona, porém isso dificultaria a sua análise de estabilidade.

A Figura 9 representa uma implementação da arquitetura proposta para o controle da navegação de um robô móvel em um ambiente composto por corredores, portas e salas, onde vários tipos de obstáculos podem estar presentes. Como mostra a Figura 9, cada controlador recebe informações sensoriais como entradas que, para permitir uma melhor representação do ambiente de operação do robô e uma redução do nível de ruído, podem inclusive ser o resultado de um processo de fusão sensorial. Cada controlador produz como saídas velocidades lineares e/ou angulares, que são alimentadas nas entradas dos Filtros de Informação locais. Estes, por sua vez, associados ao Filtro de Informação global, compõem um Filtro de Informação Descentralizado (vide Seção 2.4).

Na arquitetura proposta, uma covariância medindo a confiabilidade dos dados observados deve ser associada a cada Filtro de Informação local. A interpretação destas covariâncias e possíveis métodos para calculá-las serão apresentadas ainda neste capítulo. A saída do Filtro de Informação global, e portanto do FID, será mais próxima à saída do Filtro de Informação local ao qual estiver associada a menor covariância, ou seja, ao controlador que produzir as saídas mais adequadas para o cumprimento da tarefa desejada.

Comparando a arquitetura de controle proposta com as que foram mencionadas no Capítulo 1, o sistema proposto combina informações sobre velocidades lineares e angulares provenientes de diversos controladores utilizando o FID, que é um método de fusão ótimo<sup>6</sup> [1][13][19], ao passo que as outras arquiteturas mencionadas anteriormente utilizam métodos de fusão não ótimos. Além disso, ao contrário da arquitetura DAMN [32], AuRA [47], *Multiple Objective Decision Making* [39] e *Multivaluated Logic Approach* [7], a arquitetura proposta executa a fusão das velocidades lineares e angulares produzidas por cada controlador envolvido no processo diretamente, não sendo necessário, portanto, nenhum tipo de pré-processamento antes da fusão. Finalmente, somente a arquitetura proposta e a abordagem dinâmica [11] permitem uma análise formal de estabilidade.

A arquitetura proposta também apresenta um sistema supervisor, o qual é responsável por desativar controladores que estejam “fora de contexto”<sup>7</sup>. A forma como o sistema supervisor atua será descrita no Capítulo 4.

Em resumo, a arquitetura proposta é síncrona e apresenta codificação de resposta contínua, através de velocidades lineares e angulares. O método de coordenação entre os diferentes controladores consiste no Filtro de Informação Descentralizado (FID), e caso alguma análise de estabilidade do sistema seja desejável, é necessário o uso de controladores que conduzam a sistemas de controle estáveis no sentido de Lyapunov, que devem ser projetados de acordo com as conhecidas técnicas de projeto de controladores não-lineares.

### 3.1 *Interpretação das Covariâncias*

Quando se utiliza o Filtro de Informação ou o Filtro de Kalman, a covariância associada ao erro de medição é uma medida estatística da confiabilidade dos dados disponibilizados por cada fonte de informação [1].

No caso da fusão das saídas de diferentes controladores, a covariância representa uma medição do grau de adequação de um certo controlador às condições atuais do ambiente de operação do robô. Quanto menor for a covariância associada a um certo controlador, mais adequado ele é às condições de operação atuais.

Supondo que uma inferência confiável a respeito das condições do ambiente de operação está disponível, o projetista do sistema de controle pode associar, a cada instante, um grau de adequação a cada controlador. Isto é o mesmo que associar a cada controlador uma covariância correspondente.

As condições do ambiente de operação são inferidas a partir de informações provenientes do sistema de sensoriamento (que pode ou não envolver fusão sensorial), ou a partir de informações fornecidas pelo sistema de supervisão. O problema da modelagem estatística das falhas dos sensores e do ruído associado ao processo de medição é muito complexo, sobretudo quando o robô móvel é projetado para operar em ambientes variados, normalmente não estruturados. Além disso, é desejável que o robô móvel esteja apto a operar

---

<sup>6</sup> Neste caso, o conceito de ótimo refere-se ao fato de que o FID, assim como o filtro de Kalman, o erro médio quadrático entre o valor estimado e o valor medido.

<sup>7</sup> Considera-se que um controlador está fora de contexto quando não é possível medir ao menos uma de suas variáveis de estado. Por exemplo, no meio de uma sala muito ampla, o controlador de seguir corredores estará fora de contexto, pois não existe nenhum corredor a ser seguido.

em ambientes onde pessoas estejam presentes, e as ações dessas pessoas não podem ser descritas por processos determinísticos ou até mesmo estocásticos [7]. Por estas razões, modelos estatísticos dos sensores para tarefas como a navegação de robôs móveis são muito difíceis de obter **Erro! A origem da referência não foi encontrada.**, e portanto não serão considerados neste trabalho.

No entanto, o projetista do sistema pode considerar informações a respeito da confiabilidade de um certo sensor em um certo tipo de ambiente de operação ao associar uma determinada covariância a um controlador. Suponha-se por exemplo, que estão disponíveis dois controladores para executar a tarefa de navegação em corredores. O primeiro deles é baseado em informações disponibilizadas por um sistema de sensoramento ultra-sônico, e o segundo é baseado na diferença entre o fluxo ótico medido sobre a parede esquerda e a direita. Os sensores de ultra-som podem apresentar leituras incorretas por várias razões (como múltiplas reflexões e *cross-talking*), além de apresentar uma baixa resolução angular **Erro! A origem da referência não foi encontrada.**[28][31][53][54]. Por outro lado, o fluxo ótico pode ser mal interpretado devido à iluminação inapropriada do ambiente ou a insuficiência de textura da parede. Nestes casos, conhecendo os tipos de paredes e as condições de iluminação do ambiente, o projetista do sistema pode atribuir uma menor covariância (maior importância) ao controlador cuja operação se baseia em informações provenientes dos sensores mais adaptados às condições do ambiente de operação.

### 3.2 O Cálculo das Covariâncias

A covariância associada a cada controlador pode ser definida através de distintos métodos. Um deles poderia ser o uso de relações matemáticas envolvendo os dados medidos pelos sensores do robô. À medida que um controlador é mais adequado, menor deve ser o resultado da equação que define a sua covariância.

No entanto, um método mais simples de calcular a covariância consiste na utilização de lógica *fuzzy*. Neste caso, algumas variáveis linguísticas são utilizadas para representar o conhecimento que o projetista possui a respeito do sistema de controle do robô e de seu ambiente de operação, através de uma base de regras. Esta é a solução adotada neste trabalho, para a qual três variáveis *fuzzy* são utilizadas como antecedentes. Elas são a menor distância medida pelos sensores de ultra-som frontais do robô ( $d_{min}$ ), o produto das distâncias medidas pelos sensores de ultra-som à direita ( $d_{dir}$ ) e à esquerda ( $d_{esq}$ ) do robô, e o menor dos valores entre  $d_{dir}$  e  $d_{esq}$ . Os conjuntos nebulosos e as funções de pertinência para estes três antecedentes são idênticos, e são apresentados na Figura 10. Os consequentes são as covariâncias associadas a cada um dos quatro controladores. Elas são  $R_1$  para o controlador responsável por executar a tarefa de navegar em corredores,  $R_2$  para o controlador de Posição Final,  $R_3$  para o controlador responsável por evitar obstáculos e  $R_4$  para o controlador responsável por seguir paredes (vide Figura 9). Os consequentes são modelados como *singletons* [43], ou seja, como números reais.

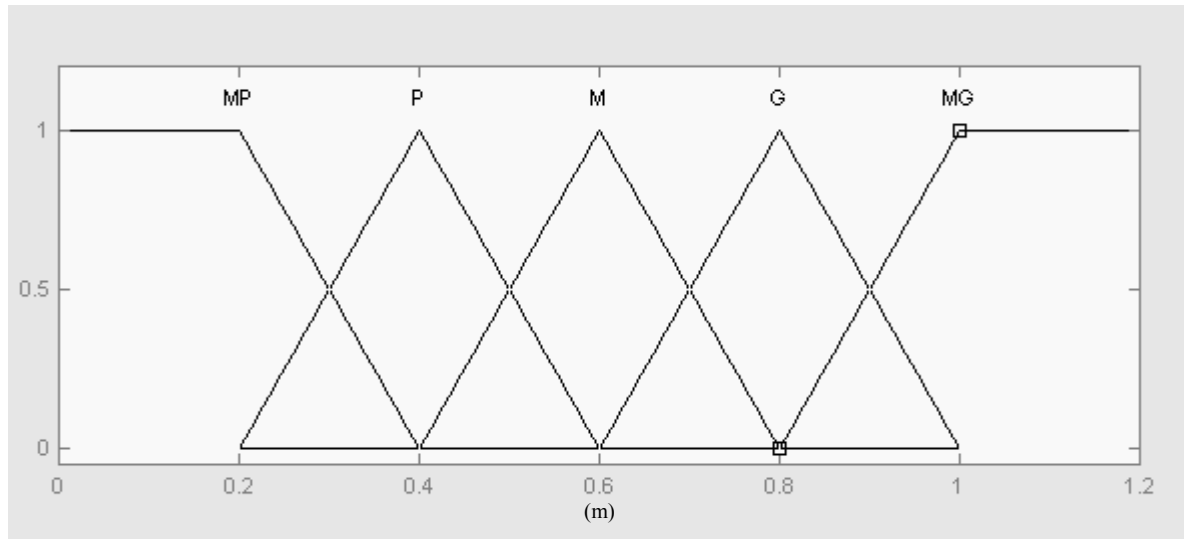


Figura 10 Funções de pertinência das variáveis de entrada (antecedentes).

A *defuzzyficação* é realizada utilizando o método da média *fuzzy* [43]. Desse modo, o valor real a ser atribuído à covariância correspondente,  $R_i$ , é dado por:

$$R_i = \frac{\sum_{i=1}^k \beta_i b_i}{\sum_{i=1}^k \beta_i} \quad (25)$$

onde  $b_i$  são os valores *singleton* e  $\beta_i$  são os graus de pertinência dos antecedentes.

A Tabela 5 e a Tabela 5 mostram as regras *fuzzy* adotadas para determinar a covariância atribuída a cada controlador, onde MP significa *muito pequeno*, P significa *pequeno*, M significa *médio*, G significa *grande* e MG significa *muito grande*. Na implementação da arquitetura apresentada na Figura 9, foram considerados os seguintes valores: MP = 1, P = 8, M = 27, G = 64 e MG = 125.

Tabela 4 Regras *fuzzy* utilizadas para determinar a covariância do controlador responsável por executar a tarefa de navegar por corredores.

$d_{esq} \times d_{dir}$	<b>MP</b>	<b>P</b>	<b>M</b>	<b>G</b>	<b>MG</b>
<b><math>d_{min}</math></b>					
<b>MP</b>	$R_l=MG$	$R_l=MG$	$R_l=MG$	$R_l=MG$	$R_l=MG$
<b>P</b>	$R_l=P$	$R_l=M$	$R_l=G$	$R_l=MG$	$R_l=MG$
<b>M</b>	$R_l=MP$	$R_l=MP$	$R_l=MP$	$R_l=MG$	$R_l=MG$
<b>G</b>	$R_l=MP$	$R_l=MP$	$R_l=MP$	$R_l=MG$	$R_l=MG$
<b>MG</b>	$R_l=MP$	$R_l=MP$	$R_l=MP$	$R_l=MG$	$R_l=MG$

Tabela 5 Regras *fuzzys* utilizadas para determinar as covariâncias dos controladores responsáveis por executar as tarefas de buscar a posição final ( $R_2$ ), evitar obstáculos ( $R_3$ ) e seguir paredes ( $R_4$ ).

$Min(d_{esq}, d_{dir})$ $d_{min}$	$MP$	$P$	$M$	$G$	$MG$
$MP$	$R_2=MG$ $R_3=MP$ $R_4=MG$	$R_2=MG$ $R_3=MP$ $R_4=MG$	$R_2=MG$ $R_3=MP$ $R_4=MG$	$R_2=MG$ $R_3=MP$ $R_4=MG$	$R_2=MG$ $R_3=MP$ $R_4=MG$
$P$	$R_2=MG$ $R_3=MP$ $R_4=P$	$R_2=MG$ $R_3=MP$ $R_4=M$	$R_2=MG$ $R_3=MP$ $R_4=G$	$R_2=MG$ $R_3=MP$ $R_4=MG$	$R_2=MG$ $R_3=MP$ $R_4=MG$
$M$	$R_2=MG$ $R_3=MG$ $R_4=MP$	$R_2=MG$ $R_3=MG$ $R_4=MP$	$R_2=MG$ $R_3=MG$ $R_4=MP$	$R_2=MG$ $R_3=MG$ $R_4=MG$	$R_2=MG$ $R_3=MG$ $R_4=MG$
$G$	$R_2=MG$ $R_3=MG$ $R_4=MP$	$R_2=MG$ $R_3=MG$ $R_4=MP$	$R_2=G$ $R_3=MG$ $R_4=MP$	$R_2=P$ $R_3=MG$ $R_4=MG$	$R_2=MP$ $R_3=MG$ $R_4=MG$
$MG$	$R_2=MG$ $R_3=MG$ $R_4=MP$	$R_2=MG$ $R_3=MG$ $R_4=MP$	$R_2=G$ $R_3=MG$ $R_4=MP$	$R_2=P$ $R_3=MG$ $R_4=MG$	$R_2=MP$ $R_3=MG$ $R_4=MG$

Como uma ilustração de tais regras, suponha que  $d_{min}$  é pequeno, o que significa um grande risco de colisão. Então, uma covariância pequena deve ser atribuída ao controlador responsável por evitar obstáculos, enquanto que covariâncias com valores elevados devem ser atribuídas aos outros controladores. Caso contrário, uma covariância com valor elevado deve ser atribuída ao controlador responsável por evitar obstáculos. Se, além disso, o produto  $d_{esq} \times d_{dir}$  for pequeno, isto significa que o robô está passando por um corredor. Portanto, o controlador responsável por seguir corredores é que deve ter a menor covariância associada a ele. Se  $d_{min}$  e o produto  $d_{esq} \times d_{dir}$  não são pequenos, mas o valor de  $d_{esq}$  ou  $d_{dir}$  é pequeno, é porque existe uma parede à esquerda ou à direita do robô. Neste caso, o controlador responsável por seguir paredes é que deve ter a menor covariância associada a ele. Finalmente, caso  $d_{min}$ ,  $d_{esq}$ ,  $d_{dir}$  e o produto  $d_{esq} \times d_{dir}$  não tenham valores pequenos, o robô provavelmente estará em um ambiente aberto, e então pode buscar o seu ponto de destino. Para assegurar que isso ocorra, o menor valor de covariância deve ser atribuído ao controlador de posição final.

## Capítulo 4

### Análise de Estabilidade

Neste capítulo é apresentada uma análise de estabilidade para a arquitetura de controle apresentada neste trabalho. São considerados dois casos. No primeiro deles os diferentes controladores envolvidos no processo de fusão têm o mesmo objetivo de controle. No segundo caso, os controladores envolvidos no processo de fusão possuem objetivos de controle diferentes.

Para projetar um sistema de controle baseado na nova arquitetura proposta (Figura 9) é necessário atender a alguns requerimentos. Inicialmente, é necessário garantir que o sistema preenche algumas condições para que apresente um “bom comportamento”, as quais devem ser expressas como condições de estabilidade. Como parte destas condições, todos os controladores utilizados devem conduzir a sistemas de controle estáveis no sentido de Lyapunov, o que permite a atribuição de uma função energia a cada um deles (funções de Lyapunov normalizadas). No Capítulo 5, os quatro controladores utilizados no sistema de controle implementado (vide Figura 9) são apresentados, e para cada um deles, é definida a função energia associada, sendo a energia do sistema o somatório das funções energia associadas aos controladores ativos em cada momento.

A função energia do sistema é dada pela seguinte equação:

$$V = \sum_i S_i V_i \quad (26)$$

onde  $V$  é a função energia do sistema,  $V_i$  é a função energia associada ao  $i$ -ésimo controlador e  $S_i$  representa o estado do controlador (ativo –  $S_i = 1$ , ou inativo –  $S_i = 0$ ), determinado pelo sistema supervisor com o objetivo de eliminar os controladores que estejam fora de contexto.

Em segundo lugar, o ambiente de operação do robô móvel deve ser parcialmente estruturado. Isto significa que o robô não tem nenhuma informação a respeito dele, exceto por informações muito gerais, como as exigências de que o ambiente seja fechado e plano. Também é necessário garantir que o ponto de destino esteja localizado em uma área livre do ambiente.

A respeito do aspecto de estabilidade, são considerados dois casos de navegação com relação à arquitetura de controle proposta. No primeiro caso, os controladores ativos em uma certa etapa de navegação são tais que eles possuem o mesmo objetivo de controle. No segundo caso, os objetivos de controle dos controladores ativos são diferentes. Ambos os casos são analisados nas subseções seguintes. Para o primeiro caso é comprovado que o sistema possui uma solução *ultimately bounded*, enquanto que para o segundo caso é proposta uma conjectura para assegurar o “bom comportamento” do sistema, baseada em funções energia. O responsável por garantir o cumprimento desta conjectura é o sistema supervisor, mostrado na Figura 9.

#### 4.1 Fusão das Saídas de Vários Controladores com o Mesmo Objetivo de Controle

Nesta Seção é realizada uma análise de estabilidade do sistema de controle resultante da fusão das saídas de diferentes controladores com um mesmo objetivo de controle. Por exemplo, suponha que estão disponíveis três controladores para executar a tarefa de navegar ao longo de corredores. O primeiro controlador é baseado em informações provenientes do sistema de sensoramento ultra-sônico, que informa ao robô a sua posição e orientação relativa ao eixo central do corredor. O segundo controlador tenta igualar o fluxo óptico medido nas paredes à direita e à esquerda do robô. O último controlador tenta igualar os ângulos formados pela junção das paredes com o piso no plano da imagem. Cada um destes controladores produz como saída um sinal de controle que representa o valor desejado da velocidade angular, e, individualmente, cada um deles deve conduzir a um sistema de controle estável no sentido de Lyapunov, ao menos quando a demonstração de estabilidade do sistema leva em consideração o modelo cinemático do robô.

Como um primeiro passo, considere que somente um controlador está sendo utilizado, como ilustra a Figura 11.

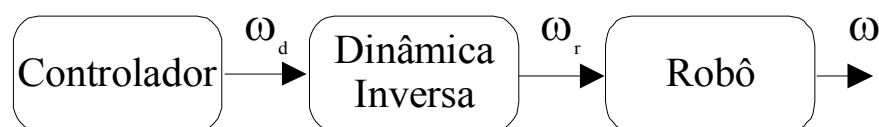


Figura 11 Sistema com somente um controlador.

Sejam:

- $\omega_d$  a velocidade angular desejada;
- $\omega$  a velocidade angular medida;
- e  $\omega_r$  a velocidade angular que deve ser enviada aos motores do robô para que apesar da dinâmica do robô,  $\omega = \omega_d$ .



Considerando que a dinâmica da velocidade angular do robô pode ser modelada como

$$\frac{\omega}{\omega_r} = \frac{k}{s^2 + as + b}$$

de maneira que  $\omega_r$  pode ser calculado como

$$\omega_r = \frac{1}{k}(\ddot{\omega} + a\dot{\omega} + b\omega) \quad (27)$$

Utilizando uma lei de controle de dinâmica inversa [29] dada por

$$\omega_r = \frac{1}{k}(\eta + a\dot{\omega} + b\omega)$$

onde

$$\eta = \ddot{\omega}_d + k_d \dot{\tilde{\omega}} + k_p \tilde{\omega} \quad k_p, k_d > 0 \quad (28)$$

$$\tilde{\omega} = \omega_d - \omega$$

A equação de laço fechado para o conhecimento exato da dinâmica do robô é dada por

$$\eta = \ddot{\omega}$$

Então, substituindo a lei de controle de (28) é possível obter

$$\ddot{\tilde{\omega}} + k_d \dot{\tilde{\omega}} + k_p \tilde{\omega} = 0$$

o que implica  $\tilde{\omega}(t) \rightarrow 0$  quando  $t \rightarrow \infty$ .

Porém, se forem utilizados mais de um controlador com o mesmo objetivo de controle, como ilustra a Figura 12, e supondo que todas as variáveis de estado associadas a eles estão disponíveis em todos os instantes de tempo, é possível escrever o seguinte conjunto de sinais de controle

$$\begin{aligned} \omega_{r1} &= \frac{1}{k}(\eta_1 + a\dot{\omega} + b\omega) \\ \omega_{r2} &= \frac{1}{k}(\eta_2 + a\dot{\omega} + b\omega) \\ &\vdots \\ \omega_{rn} &= \frac{1}{k}(\eta_n + a\dot{\omega} + b\omega) \end{aligned}$$

Então, o sinal de controle fusionado é

$$\hat{\omega}_r = \frac{1}{k} (\hat{\eta} + a\dot{\omega} + b\omega) \quad (29)$$

Um sinal de controle ideal  $\omega_d = \omega_{di} + \Delta\omega_{di}$  corresponde a um  $\eta$  ideal tal que

$$\begin{aligned} \eta &= \eta_1 + \Delta\eta_1 \\ \eta &= \eta_2 + \Delta\eta_2 \\ &\vdots \\ \eta &= \eta_n + \Delta\eta_n \end{aligned}$$

o que resulta em

$$\eta = \hat{\eta} + \Delta\hat{\eta} \quad (30)$$

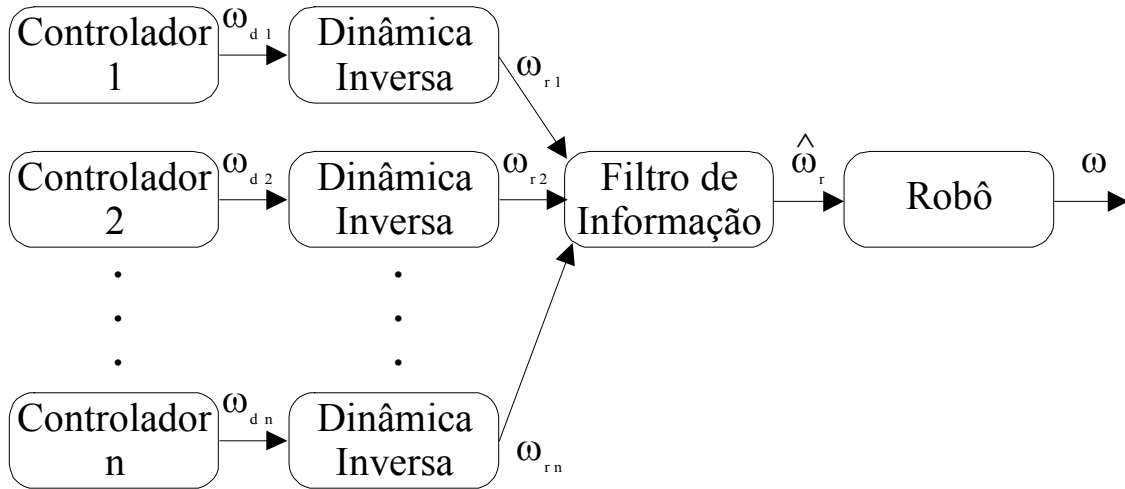


Figura 12 Fusão das saídas de diferentes controladores.

Mais uma vez, a equação de laço fechado para o conhecimento exato da dinâmica do robô é dada por

$$\hat{\eta} = \ddot{\omega}$$

e, finalmente, levando em conta (30)

$$\hat{\eta} = \eta - \Delta\hat{\eta} = \ddot{\omega} \quad (31)$$

Para (28) e (31) é possível escrever a seguinte dinâmica para o erro da velocidade angular

$$\ddot{\tilde{\omega}} + k_d \dot{\tilde{\omega}} + k_p \tilde{\omega} = \Delta\hat{\eta} \quad (32)$$

Definindo o vetor de estados  $\mathbf{x} = [\tilde{\omega} \quad \dot{\tilde{\omega}}]^T$ , (32) pode ser escrita como

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \boldsymbol{\delta}(\mathbf{x}) \quad (33)$$

onde:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -k_p & -k_d \end{pmatrix} \quad \boldsymbol{\delta}(\mathbf{x}) = \begin{pmatrix} 0 \\ \Delta \hat{\eta} \end{pmatrix}$$

Pode-se provar então que o sistema descrito por (33) possui uma solução *ultimately bounded solution* [23]. Isto significa que existem  $b, c > 0$ , tais que para cada  $\alpha \in (0, c)$  existe uma constante positiva  $T = T(\alpha)$  tal que:

$$\|\mathbf{x}(t_0)\| < \alpha \Rightarrow \|\mathbf{x}(t)\| \leq b \quad \forall t \geq t_0 + T(\alpha)$$

onde  $b$  é a cota final.

Tomando a seguinte função candidata de Lyapunov

$$V = \mathbf{x}^T \mathbf{P} \mathbf{x}, \quad \mathbf{P} = \mathbf{P}^T > 0$$

sua derivada temporal é

$$\dot{V} = -\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2\mathbf{x}^T \mathbf{P} \boldsymbol{\delta}(\mathbf{x}) \quad (34)$$

onde

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q}$$

Tomando limites para os dois termos de (34)

$$-\mathbf{x}^T \mathbf{Q} \mathbf{x} \leq -\lambda_{\min}(\mathbf{Q}) \|\mathbf{x}\|^2$$

$$\|2\mathbf{x}^T \mathbf{P}\| \leq 2\lambda_{\max}(\mathbf{P}) \|\mathbf{x}\|$$

é possível escrever

$$\dot{V} \leq -\lambda_{\min}(\mathbf{Q}) \|\mathbf{x}\|^2 + 2\lambda_{\max}(\mathbf{P}) \|\mathbf{x}\| \|\boldsymbol{\delta}(\mathbf{x})\| \quad (35)$$

A partir de (33)

$$\|\boldsymbol{\delta}(\mathbf{x})\| \leq |\Delta \hat{\eta}|$$

De acordo com (35) é possível escrever

$$\dot{V} \leq -(1-\theta)\lambda_{\min}(\mathbf{Q}) \|\mathbf{x}\|^2 - \theta\lambda_{\min}(\mathbf{Q}) \|\mathbf{x}\|^2 + 2\lambda_{\max}(\mathbf{P}) \|\mathbf{x}\| \Delta \hat{\eta}$$

com  $0 < \theta < 1$ . Finalmente obtém-se

$$\dot{V} \leq -(1-\theta)\lambda_{\min}(\mathbf{Q})\|\mathbf{x}\|^2, \forall \|\mathbf{x}\| \geq \frac{2\lambda_{\max}(\mathbf{P})|\Delta\hat{\eta}|}{\lambda_{\min}(\mathbf{Q})\theta}$$

de maneira que a cota final [23] é

$$b = \frac{2\lambda_{\max}(\mathbf{P})}{\lambda_{\min}(\mathbf{Q})} \sqrt{\frac{\lambda_{\max}(\mathbf{P})}{\lambda_{\min}(\mathbf{P})}} \frac{|\Delta\hat{\eta}|}{\theta}$$

Como a fusão dos sinais de controle é realizada utilizando um Filtro de Informação Descentralizado (que, assim como o Filtro de Kalman, é um método de fusão ótimo [1]), a cota final do desvio padrão do erro final é menor do que aquelas que correspondem aos erros produzidos por cada controlador. Isto significa que o erro entre o valor real e o valor desejado da velocidade angular converge a uma região limitada em um tempo finito (*ultimately bounded solution*), e que a dimensão de tal região é menor do que quando se utiliza separadamente qualquer um dos controladores envolvidos no processo de fusão.

O caso mais abrangente, onde os controladores envolvidos no processo de fusão possuem objetivos de controle diferentes, é abordado na próxima seção.

#### 4.2 Fusão das Saídas de Vários Controladores com Objetivos de Controle Diferentes

A análise de estabilidade realizada na seção anterior não tem validade quando os controladores envolvidos no processo de fusão não possuem os mesmos objetivos de controle. Um exemplo é o sistema mostrado na Figura 9, no qual os quatros controladores envolvidos possuem objetivos de controle diferentes. Neste caso, propõe-se uma conjectura baseada em fases de navegação e energia associada ao sistema.

Quando o robô está navegando entre um ponto de partida e um ponto de destino (objetivo final) ele passa por diversas fases de navegação. Uma fase de navegação é uma parte do caminho seguido pelo robô onde somente um objetivo de controle predomina. Se o objetivo principal de controle muda, uma fase de navegação termina e outra se inicia. O sistema de controle detecta que ocorreu uma mudança de fase de navegação quando a função energia associada a pelo menos um dos controladores cresce mais rápido do que cresceria em condições normais (devido ao fato de que tanto a velocidade angular quanto a linear são limitadas). Este tipo de crescimento é denominado de abrupto, enquanto que um crescimento normal se denomina gradual. Alguns exemplos de fases de navegação são seguir paredes, evitar obstáculos, seguir corredores, etc.

Desse modo, um detalhe importante ao se projetar um sistema de controle utilizando a arquitetura proposta é que deve existir pelo menos um controlador adequado a cada fase de navegação com que o robô irá se deparar ao longo do trajeto.

A respeito da estabilidade do sistema quando no processo de fusão estão envolvidos controladores com objetivos de controle diferentes, supõe-se que a função energia do sistema deve decrescer enquanto o robô permanece em uma fase de navegação. Para garantir isso, um sistema supervisor (ou simplesmente um supervisor) é adicionado à arquitetura de controle (conforme ilustra a Figura 9) para monitorar a função energia associada a cada controlador e também a função energia do sistema global. Então, se a função energia do sistema começa a crescer gradualmente, os controladores cujas funções energia estejam crescendo gradualmente serão eliminados do processo de fusão pelo supervisor, ou seja, serão desativados (o que é equivalente a fazer a covariância associada a eles igual a infinito).

Como o ambiente de operação é desconhecido, os tipos e número de fases de navegação ao longo do trajeto do robô são desconhecidos. Também não é possível determinar o momento exato em que uma mudança de fase de navegação irá ocorrer. Devido a isso, é possível considerar a mudança de fase de navegação como sendo uma perturbação. Por essa razão, permite-se que a função energia do sistema cresça na transição entre duas fases de navegação.

O supervisor também deve eliminar do processo de fusão os controladores que estiverem “fora de contexto”. Diz-se que um controlador está fora de contexto quando as suas variáveis de estado não estão disponíveis. Um exemplo desta situação é quando o robô está no centro de uma sala muito grande. Como neste caso o seu sistema de sensoramento (neste trabalho, somente sensores ultra-sônicos) não detecta nenhuma parede, o controlador de seguir paredes e o controlador de seguir corredores estão ambos fora de contexto, pois não podem operar, já que o robô não consegue detectar uma parede ou um corredor para seguir. Para validar a conjectura acima referida, vários experimentos foram realizados e são apresentados no Capítulo 6.

Na próxima seção, o funcionamento do sistema supervisor aqui desenvolvido é descrito em detalhes.

### **4.3 Sistema Supervisor**

O sistema supervisor implementado para garantir o cumprimento da conjectura para que o sistema apresente um “bom comportamento” é mostrado na Figura 13.

Para atingir este objetivo, o sistema supervisor necessita executar o seguinte conjunto de operações:

- Monitorar a função energia do sistema e as funções energia dos controladores utilizados.
- Caso a função energia do sistema apresente um crescimento gradual, o sistema supervisor deve eliminar do processo de fusão os controladores cujas funções energia associadas também apresentem um crescimento gradual.
- Eliminar do processo de fusão aqueles controladores que estiverem “fora de contexto”.
- Verificar se houve mudança na fase de navegação, e em caso positivo, assinalar.

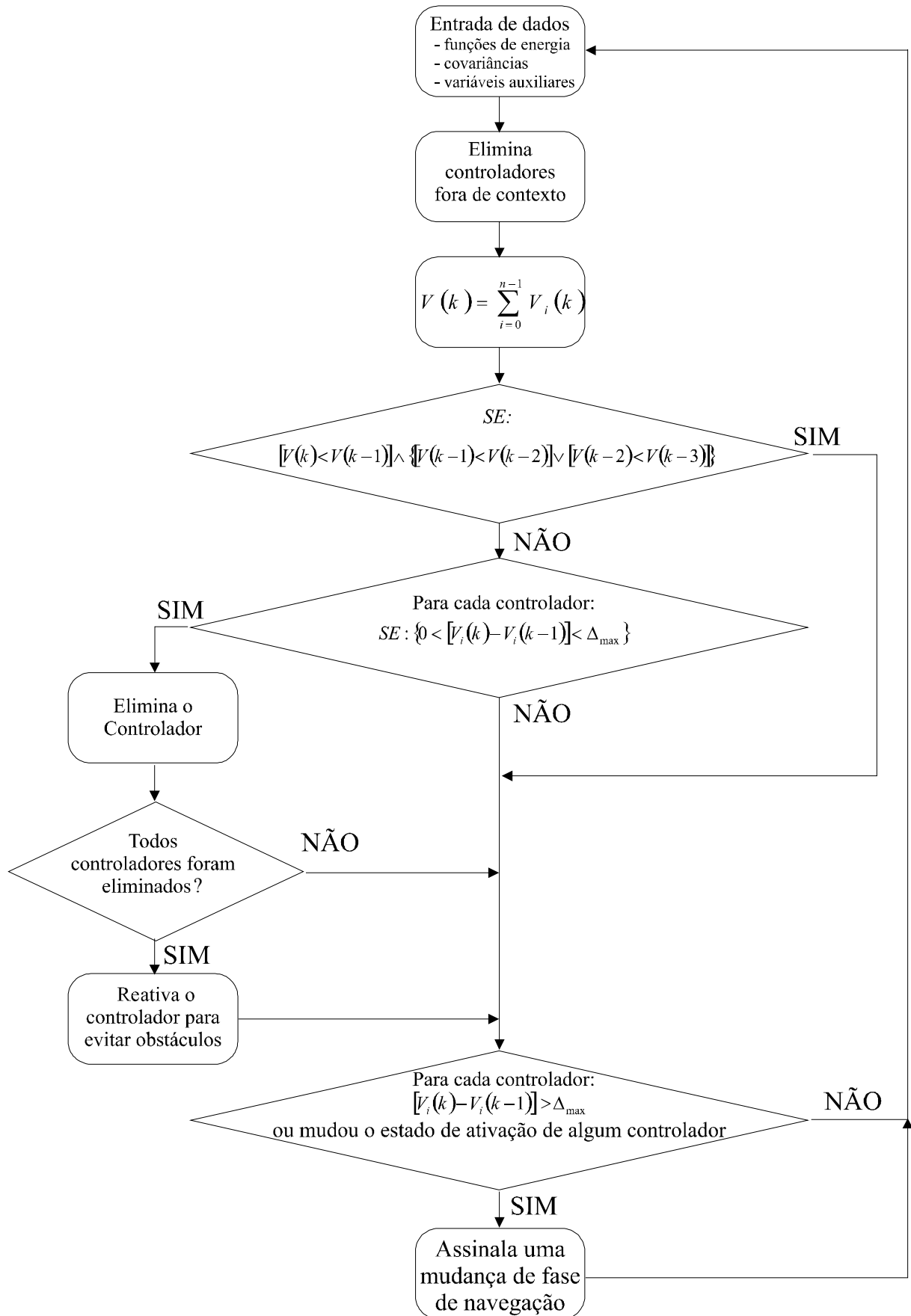


Figura 13 Sistema Supervisor implementado para garantir o cumprimento da conjectura para que o sistema apresente um “bom comportamento”.

Para o seu funcionamento, o sistema supervisor necessita ser alimentado com informações provenientes de diversas partes do sistema de controle para que possa atuar adequadamente, conforme ilustra a Figura 9. Como dados de entrada, o supervisor necessita das funções energia associada a cada controlador, das covariâncias associadas a cada controlador e de algumas variáveis auxiliares, dentre elas algumas das variáveis utilizadas como antecedentes para a determinação através de lógica *fuzzy* das covariâncias associadas aos controladores, são elas:  $d_{esq}$  (43),  $d_{der}$  (44),  $\alpha$  (Figura 14). Com base nestes valores, são determinadas outras variáveis auxiliares que servem também para avaliar se os controladores estão em contexto.

#### 4.3.1 Determinação dos Controladores que Estão Fora de Contexto

O controlador de Posição Final está quase sempre em contexto. O único momento em que ele fica fora de contexto é quando o robô móvel se aproxima do seu ponto de destino a uma distância menor que um determinado limiar. Este é também o critério de parada do robô móvel. Devido a erros odométricos, não se pode esperar que o robô móvel chegue exatamente no ponto previsto. Assim, foi definido um limiar variável, que é função da distância percorrida pelo robô (já que os erros odométricos são acumulativos), conforme a equação

$$LP = 0,1 + \frac{DP}{50}$$

onde  $LP$  é o limiar de parada e  $DP$  é a distância percorrida, em metros, calculada com base nos dados odométricos do robô móvel. Nas simulações e experimentos apresentados no Capítulo 6, os valores de  $DP$  variam de 5 a 17,5 metros, e, portanto, o limiar de parada ( $LP$ ) varia de 20 a 45 cm. Quando a distância do robô ao ponto de destino é menor do que  $LP$ , o critério de parada é satisfeito, e o controlador de Posição Final ficaria fora de contexto, se o experimento ou simulação não fosse encerrado.

Quanto ao controlador para evitar obstáculos, ele fica fora de contexto sempre que nenhum obstáculo é detectado pelo sistema de sensoriamento do robô. Porém, como o algoritmo implementado para evitar obstáculos somente apresenta alguma contribuição para a navegação do robô móvel quando ao menos um obstáculo é detectado, permanecendo inerte todo tempo restante, não é necessário que o sistema supervisor se encarregue de verificar se ele está ou não em contexto.

O controlador de seguir corredores somente está dentro de contexto se forem detectadas paredes a uma distância menor que 2,5 metros em ambos os lados do robô. Para isso é que são utilizadas as variáveis  $d_{esq}$  e  $d_{dir}$ .

O controlador para seguir paredes somente está em contexto se houver uma parede à esquerda ou à direita do robô. Caso existam paredes em ambos os lados, este controlador está fora de contexto, pois o controlador que deve atuar neste caso é o controlador de seguir corredores, e o robô não pode posicionar-se no centro de um corredor e a uma certa distância fixa de uma das duas paredes ao mesmo tempo, a não ser em uma situação especial onde o centro do corredor está à distância que se deseja seguir a parede. Outro aspecto que o sistema supervisor analisa, para verificar se o controlador de seguir paredes está em contexto, é a direção em que se encontra o ponto de destino. Por exemplo, se a parede que está sendo seguida está à direita do robô e o ponto de destino está à esquerda, o sistema supervisor

considera o controlador de seguir paredes como estando fora de contexto, pois o objetivo primário do robô é buscar o ponto de destino e não seguir paredes. Por isso é que é necessário informar ao sistema supervisor o ângulo  $\alpha$  que indica a direção que o robô deve seguir para encontrar o ponto de destino.

#### 4.3.2 Determinação de Quais Controladores Devem Ser Eliminados do Processo de Fusão

Após o sistema supervisor determinar quais controladores estão fora de contexto e elimina-los do processo de fusão, ele passa a verificar se a função energia do sistema está decrescendo ou não. Caso a função energia do sistema esteja decrescendo, o sistema supervisor não toma nenhuma atitude. Caso ocorra um crescimento na função energia do sistema, o supervisor verificará o comportamento da função energia associada a cada controlador e verificará quais apresentaram crescimento e determinará se o crescimento foi gradual ou abrupto<sup>8</sup>. Considera-se que a função energia associada a um controlador sofreu um aumento abrupto se o aumento exceder um valor  $\Delta_{\max}$  que é determinado de acordo com as limitações físicas de aceleração linear e angular do robô móvel. A crescimentos abruptos o supervisor associa uma mudança de fase de navegação. Por outro lado, quando o aumento verificado é considerado como gradual, o sistema supervisor elimina do processo de fusão o respectivo controlador.

Como um primeiro passo para executar a tarefa de verificar se algum controlador necessita ser eliminado do processo de fusão por não estar contribuindo positivamente para a realização dos objetivos de controle (o que se reflete no crescimento gradual da sua função energia associada), o supervisor calcula a função energia do sistema (26). O supervisor também mantém os três últimos valores da função energia do sistema, e verifica se houve aumento através do seguinte teste

$$[V(k) < V(k-1)] \wedge \{ [V(k-1) < V(k-2)] \vee [V(k-2) < V(k-3)] \} \quad (36)$$

Pode-se verificar por (36) que, para considerar que houve um aumento na função energia do sistema, o supervisor exige que, além de ter ocorrido um aumento da função energia do sistema no instante  $t_k$  com relação ao instante  $t_{k-1}$ , é necessário que, entre os instantes  $t_{k-1}$  e  $t_{k-2}$  ou  $t_{k-2}$  e  $t_{k-3}$ , também tenha ocorrido um aumento da função energia do sistema. Isto é feito para que o supervisor reaja a uma tendência de aumento da função energia do sistema, e não a um aumento pontual e isolado, que pode ser devido a um ruído dos sensores que prejudicou a medição das variáveis de estado utilizadas nos controladores e, baseada nas quais, é calculada a função energia associada a cada controlador. Caso não tenha de fato havido um aumento na energia do sistema, o supervisor passará a verificar se houve uma mudança de fase de navegação.

---

<sup>8</sup> Um crescimento é considerado abrupto quando ele excede o maior crescimento que seria possível se as velocidades linear e angular sofressem a variação mais brusca possível (devido às suas limitações físicas) em uma unidade de tempo, que no caso do robô móvel PIONEER 2DX utilizado nas simulações e experimentos corresponde a um tempo de 100ms.



Caso tenha ocorrido realmente um aumento na função energia do sistema, o sistema supervisor passará a verificar, para cada controlador, se a sua função energia sofreu algum aumento, e, em caso positivo, se foi um aumento gradual ou abrupto. Para isso, o sistema supervisor realiza, para cada controlador, o seguinte teste

$$0 < [V_i(k) - V_i(k-1)] < \Delta_{\max} \quad (37)$$

O teste de (37) verifica inicialmente se houve de fato um aumento na função energia associada ao controlador em questão. Para isso, o sistema supervisor compara o último valor da função energia recebido com o valor anterior (que o sistema supervisor necessita manter armazenado). Se houve decréscimo ( $[V_i(k) - V_i(k-1)] < 0$ ), então este controlador está contribuindo positivamente para atingir os objetivos de controle, e, portanto, não deve ser tocado. Por outro lado, se  $[V_i(k) - V_i(k-1)] > \Delta_{\max}$  significa que houve um aumento abrupto e portanto considera-se que ocorreu uma mudança de fase de navegação. No caso em que  $0 < [V_i(k) - V_i(k-1)] < \Delta_{\max}$ , houve um aumento gradual na função energia do sistema, e portanto o controlador em questão deve ser eliminado do processo de fusão.

Caso venha a ocorrer que as funções energia de todos controladores apresentem um crescimento gradual, todos eles seriam eliminados do processo de fusão, e o robô não teria nenhum controlador ativo. Por mais improvável que isso seja, o supervisor toma o cuidado de verificar, e em caso positivo, ele reativa o controlador de evitar obstáculos, que é o mais importante para manter a integridade física do robô.

#### 4.3.3 Determinação da Ocorrência de Mudança de Fase de Navegação

Normalmente, uma mudança de fase de navegação é identificada pela ocorrência de um aumento abrupto na função energia associada a um ou mais controladores, conforme a seguinte equação

$$[V_i(k) - V_i(k-1)] > \Delta_{\max}$$


onde  $V_i(k)$  é a função energia associada ao  $i$ -ésimo controlador, medida no instante  $t_k$ .

Porém, pode existir ainda um caso em que nenhuma das funções energia associadas aos controladores apresente um crescimento abrupto, e mesmo assim seja necessário considerar que houve uma mudança de fase de navegação. Isto ocorre quando o estado de ativação de ao menos um dos controladores mudou. Por exemplo, se o estado de ativação do controlador de seguir paredes estava inativo e passou para ativo, significa que o robô entrou em um corredor, e, logo, houve uma mudança de fase de navegação. Conforme demonstram as simulações e experimentos apresentados no Capítulo 6, todos os aumentos na função energia do sistema correspondem a pontos onde ocorreram uma mudança de fase de navegação, comprovando a conjectura proposta nesta seção para o “bom comportamento do sistema”.

Caso fosse necessário incluir no sistema um componente deliberativo, este poderia tomar o lugar do sistema supervisor. Tal componente deliberativo poderia ser tão complexo como fosse necessário, podendo incluir um ou mais módulos, e poderia ser assíncrono com relação ao resto do sistema de controle. A contribuição do componente deliberativo ao sistema poderia se dar de várias maneiras. Uma delas seria como atua o sistema supervisor descrito neste capítulo. Além disso, o componente deliberativo poderia calcular as covariâncias associadas aos controladores, definir grupos de controladores que seriam ativados para cumprir certas tarefas, e também, a partir de algum mapa do ambiente de operação do robô, tal componente deliberativo poderia enviar para o sistema de controle reativo pontos de destino intermediários, o que facilitaria a tarefa de navegação do robô móvel.

## Capítulo 5

# Controladores Utilizados



Neste capítulo são apresentados os quatro controladores utilizados neste trabalho (vide Figura 9). Inicialmente, será apresentado o controlador para navegação em corredores e, em seguida, como ele pode ser adaptado para executar a tarefa de seguir paredes. Em seguida, apresenta-se o controlador de posição final [24]. Ao final, apresenta-se o controlador utilizado para evitar obstáculos, que consiste em uma adaptação do conhecido controlador baseado em impedância [24][25][48]. Para cada controlador apresentado neste capítulo, também será apresentada a sua função energia a ele associada.

### 5.1 *Controlador para Seguir Corredores*

Quando robôs móveis autônomos navegam em ambientes fechados, eles devem ter a capacidade de seguir corredores, paredes, dobrar esquinas, entrar e sair de salas, etc.

Para seguir corredores, alguns algoritmos baseados em visão computacional já foram propostos. Em [50] utiliza-se processamento de imagens para detectar as linhas perspectivas formadas no plano de imagem pela união das paredes com o piso para guiar o robô ao longo do eixo central do corredor. Em [57] as linhas perspectivas formadas pela união do teto com as paredes é que são utilizadas para guiar o robô. Outros autores propuseram o uso de fluxo ótico para guiar o robô ao longo de corredores. Em [33] duas câmeras de vídeo laterais montadas sobre o robô são utilizadas, e o fluxo ótico é calculado em ambas as câmeras, sendo que a diferença entre o fluxo ótico medido à direita e à esquerda é utilizada para controlar o movimento do robô. Em [8] e [49] uma câmera é utilizada para guiar o robô ao longo do eixo central do corredor, utilizando o cálculo do fluxo ótico e de suas derivadas temporais.

Nesta seção será introduzido um algoritmo de controle que permite a um robô móvel seguir o eixo central de um corredor. O algoritmo é projetado para ser assintoticamente estável e para evitar a saturação dos atuadores do robô (através da limitação dos sinais de

controle, conforme o detalhamento na Subseção 5.1.3). O algoritmo de controle proposto pode facilmente ser adaptado para permitir ao robô seguir uma parede à sua direita ou esquerda, conforme ilustra a Seção 5.2, onde o controlador aqui apresentado é adaptado para seguir paredes.

Finalmente, é importante salientar que este controlador assintoticamente estável que permite a um robô móvel seguir corredores ou paredes, consiste em uma das contribuições originais desta Tese de Doutorado.

### 5.1.1 Equações Cinemáticas

Considere o modelo tipo monociclo do robô posicionado a uma distância diferente de zero do sistema de coordenadas de destino  $\langle g \rangle$ . O movimento do robô na direção de  $\langle g \rangle$  é governado pela ação combinada tanto da velocidade angular  $\omega$  quanto da velocidade linear  $u$ , a qual está sempre na direção de um dos eixos do sistema de coordenadas  $\langle a \rangle$  que está acoplado ao robô, conforme mostra a Figura 14.

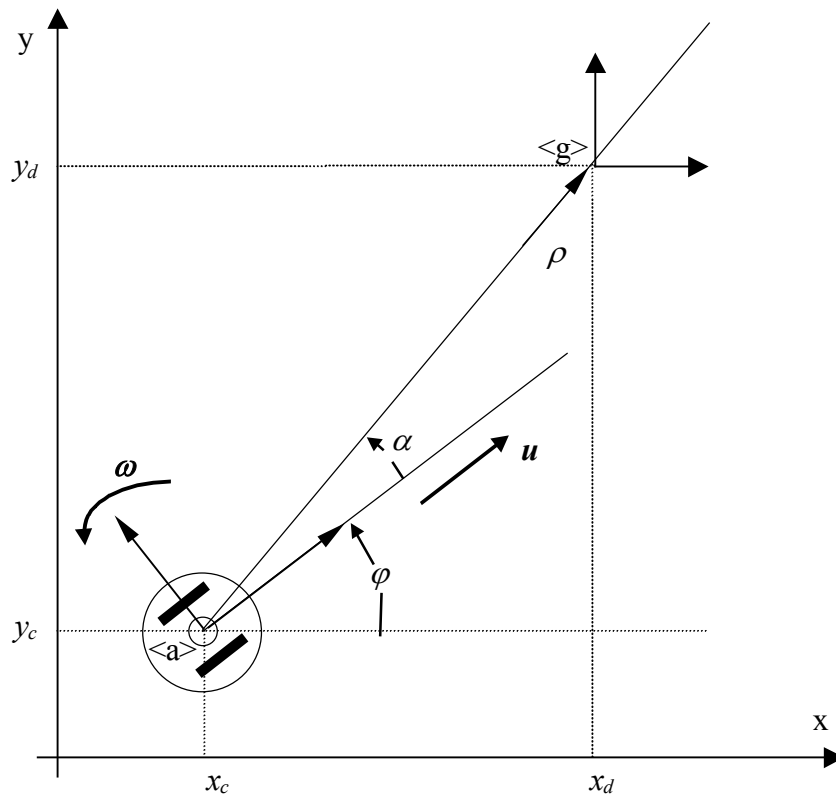


Figura 14 Posição e orientação do robô.

O conjunto usual de equações cinemáticas, as quais envolvem a posição em coordenadas cartesianas  $(x, y)$  do robô e seu ângulo de orientação  $\varphi$ , consiste em

$$\begin{cases} \dot{x} = u \cdot \cos \varphi \\ \dot{y} = u \cdot \sin \varphi \\ \dot{\varphi} = \omega \end{cases}$$

onde  $u$  é a magnitude do vetor de velocidade linear, as variáveis  $x$  e  $y$  são medidas com relação ao sistema de coordenadas de destino  $\langle g \rangle$ , e  $\varphi$  é medido com relação à orientação do eixo  $x$ .

No caso da navegação em corredores, as variáveis de estado são definidas com relação ao corredor como  $\tilde{x}$  e  $\varphi$ , onde  $\tilde{x}$  representa a distância do robô ao centro do corredor (ou a posição relativa desejada ao longo do corredor) e  $\varphi$  é o ângulo de orientação do robô com relação ao eixo de orientação do corredor, conforme ilustra a Figura 15. Neste caso, as equações cinemáticas se reduzem a

$$\begin{cases} \dot{\tilde{x}} = u \cdot \sin \varphi \\ \dot{\varphi} = \omega \end{cases} \quad (38)$$

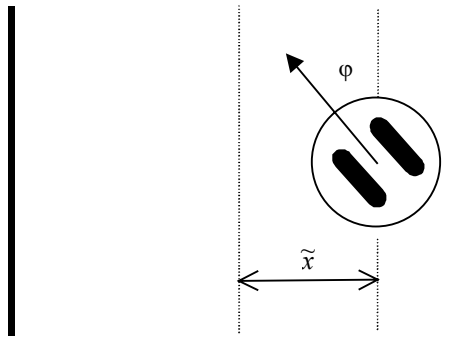


Figura 15 O robô no corredor.

### 5.1.2 Projeto do Controlador

Assumindo que o movimento de um robô móvel ao longo de um corredor pode ser descrito por (38) e que o robô se move a uma velocidade linear constante  $u$ , o objetivo de controle consiste em obter um sinal de controle  $\omega(t)$  tal que os erros de controle  $\tilde{x}(t)$  e  $\varphi(t)$  tendam assintoticamente para zero, ao mesmo tempo que o sinal de controle permanece dentro de limites especificados.

Com base nas medições das variáveis de erro  $\tilde{x}(t)$  e  $\varphi(t)$ , é proposta a seguinte lei de controle

$$\omega = -k_1(\varphi)\varphi - k_2(\tilde{x})\tilde{x}u \frac{\sin \varphi}{\varphi} \quad (39)$$

onde  $k_1(\varphi)$  e  $k_2(\tilde{x})$  são funções positivas adequadamente selecionadas que devem ser definidas para evitar a saturação da variável de controle  $\omega$ .

As equações de malha fechada correspondentes a (38) e (39) são

$$\begin{aligned}\dot{\varphi} &= -k_1(\varphi)\varphi - k_2(\tilde{x})\tilde{x}u \frac{\sin \varphi}{\varphi} \\ \dot{\tilde{x}} &= u \sin \varphi\end{aligned}\tag{40}$$

Em (40) o único ponto de equilíbrio é  $\begin{pmatrix} \tilde{x} \\ \varphi \end{pmatrix} = 0$ . Considera-se, agora, a seguinte função candidata de Lyapunov

$$V = \frac{\varphi^2}{2} + \int_0^{\tilde{x}} k_2(\eta)\eta d\eta\tag{41}$$

Esta função, definida positiva, representa a soma das energias potenciais devidas aos erros de posição e orientação do robô no corredor. Além disso,  $k_2(\tilde{x})$  é selecionada de modo que [52]:

$$k_2(x) \geq \alpha(|x|) \quad \forall x \in \mathfrak{R}$$

onde  $\alpha(\cdot)$  é uma função de classe K [35]. Então, é possível provar que o termo integral em  $V$  é radialmente não limitado. Portanto, é possível concluir que  $V$  é uma função globalmente positiva definida e radialmente não limitada.

A derivada temporal da função candidata de Lyapunov é

$$\dot{V}(\tilde{x}, \varphi) = \varphi \dot{\varphi} + k_2(\tilde{x})\tilde{x}\dot{\tilde{x}}$$

onde foi usada a regra de Leibniz para a diferenciação de integrais. Utilizando (40), obtém-se

$$\dot{V}(\tilde{x}, \varphi) = \varphi \left\{ -k_1(\varphi)\varphi - k_2(\tilde{x})\tilde{x}u \frac{\sin \varphi}{\varphi} \right\} + k_2(\tilde{x})\tilde{x}u \sin \varphi$$

$$\dot{V}(\tilde{x}, \varphi) = -k_1(\varphi)\varphi^2 \leq 0$$

que representa uma função semi-definida negativa, e daí é possível concluir que as variáveis de estado  $\tilde{x}$  e  $\varphi$  são limitadas, e  $\varphi$  é uma função temporal quadraticamente integrável.

Para provar a estabilidade assintótica, será utilizada a natureza autônoma do sistema de malha fechada representado por (40), com o objetivo de aplicar o teorema de Krasovskii-LaSalle [35]. Na região

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \varphi \end{bmatrix} : \dot{V}(\tilde{x}, \varphi) = 0 \right\} \Rightarrow \left\{ \begin{bmatrix} \tilde{x} \\ \varphi \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix} \right\}$$

o único invariante é  $\tilde{x} = 0$ , e daí, invocando o teorema de LaSalle, é possível concluir que a origem do espaço de estados é globalmente, uniformemente, assintoticamente estável.

### 5.1.3 Limitação dos Sinais de Controle

De um modo similar ao proposto em [52], a seguinte definição das funções  $k_1(\varphi)$  e  $k_2(\tilde{x})$  é considerada

$$\begin{aligned} k_1(\varphi) &= \frac{k_1}{a_1 + |\varphi|} \\ k_2(\tilde{x}) &= \frac{k_2}{a_2 + |\tilde{x}|} \end{aligned} \quad (42)$$

com  $k_1, k_2, a_1$  e  $a_2$  sendo constantes positivas. Estas funções são tais que

$$k_1(\varphi) > 0$$

$$k_2(\tilde{x}) \geq \alpha(|\tilde{x}|) \quad \forall \tilde{x} \in \mathfrak{R}$$

as quais foram consideradas na demonstração de estabilidade. Por outro lado, o valor absoluto da variável de controle  $\omega$  é

$$|\omega| \leq k_1 \left| \frac{\varphi}{a_1 + |\varphi|} \right| + k_2 \left| \frac{\tilde{x}u}{a_2 + |\tilde{x}|} \right|$$

Desde que  $\left| \frac{\tilde{x}}{a + |\tilde{x}|} \right| \leq 1$  para todo  $\tilde{x} \in \mathfrak{R}$  e  $a > 0$ , então

$$|\omega| \leq k_1 + k_2 u$$

Se forem selecionadas  $k_1$  e  $k_2$  de tal modo que  $k_1 + k_2 u \leq \omega_{\max}$  com  $|\omega| \leq \omega_{\max}$ , então o controlador garante que  $|\omega| \leq \omega_{\max}$  para todo  $t \geq 0$ .

### 5.1.4 Estimação das Variáveis de Estado

Para o funcionamento do controlador, é necessário que ele seja realimentado com os valores das variáveis de estado  $\tilde{x}(t)$  e  $\varphi(t)$  a cada momento. Estes valores podem ser inicialmente obtidos através dos dados provenientes do sistema de sensoriamento ultra-sônico do robô. A Figura 16 mostra uma situação típica onde os sensores ultra-sônicos laterais do robô móvel PIONEER 2DX<sup>9</sup>,  $S_0$ ,  $S_7$ ,  $S_8$  e  $S_{15}$ , são utilizados. As seguintes equações são então utilizadas para calcular as variáveis de estado  $\tilde{x}(t)$  e  $\varphi(t)$ .

---

<sup>9</sup> Utilizado nos experimentos e apresentado no capítulo 6.

$$d_{esq} = \frac{y_{s0} + y_{s15}}{2} \quad (43)$$

$$d_{dir} = \frac{y_{s7} + y_{s8}}{2} \quad (44)$$

$$dif = (y_{s0} - y_{s15}) = (y_{s7} - y_{s8}) \quad (45)$$

$$\tilde{x} = \frac{(d_{dir} - d_{esq})}{2}$$

$$\varphi = \text{sen}^{-1}\left(\frac{dif}{d}\right) \quad (46)$$

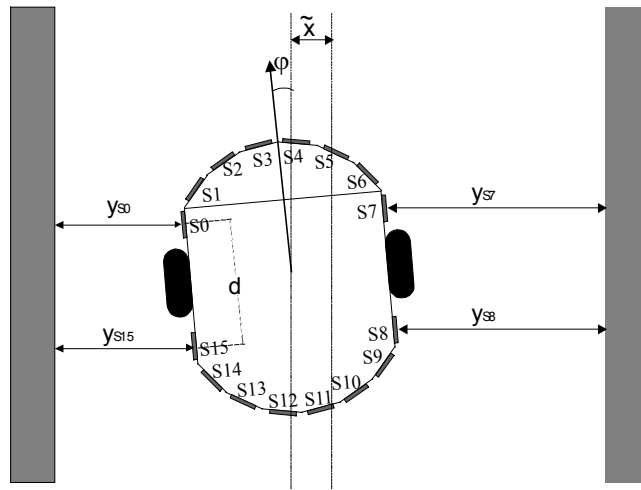


Figura 16 Variáveis de estado  $\tilde{x}(t)$  e  $\varphi(t)$  utilizadas pelo controlador projetado para seguir corredores.

Medições provenientes de sensores de ultra-som podem se deteriorar, ou até mesmo podem ser impossíveis de obter sob certas circunstâncias, como por exemplo, quando o robô está navegando e exista uma porta aberta em um corredor, ou quando o robô está orientado em um ângulo significativamente grande com relação ao eixo do corredor. Este último caso deve-se ao fato de que, um sensor de ultra-som somente está apto a coletar dados válidos, quando a orientação dele com relação ao obstáculo difere de  $90^\circ$  por um valor menor do que o semi-ângulo de abertura do lóbulo de radiação de energia do sensor. No caso do sensor eletrostático da Polaroid®, utilizado no robô móvel PIONEER 2DX, este ângulo é aproximadamente  $\varphi = 17^\circ$  [12][41][54]. Por esta razão, é importante utilizar outras fontes de informação, como por exemplo, os dados odométricos providos pelos *encoders* incrementais do robô. Alguns autores, como em [34], realizaram a fusão de dados odométricos com dados fornecidos por um sistema de sensoriamento ultra-sônico. Neste trabalho, propõe-se a fusão dos dados fornecidos pelo sistema de sensoriamento ultra-sônico com os dados resultantes de medições odométricas corrigidas, através do uso de um Filtro de Informação Descentralizado, como apresentado na Figura 17. A informação resultante do processo de fusão é, então, tomada como sendo o vetor de observação do estimador de estados.



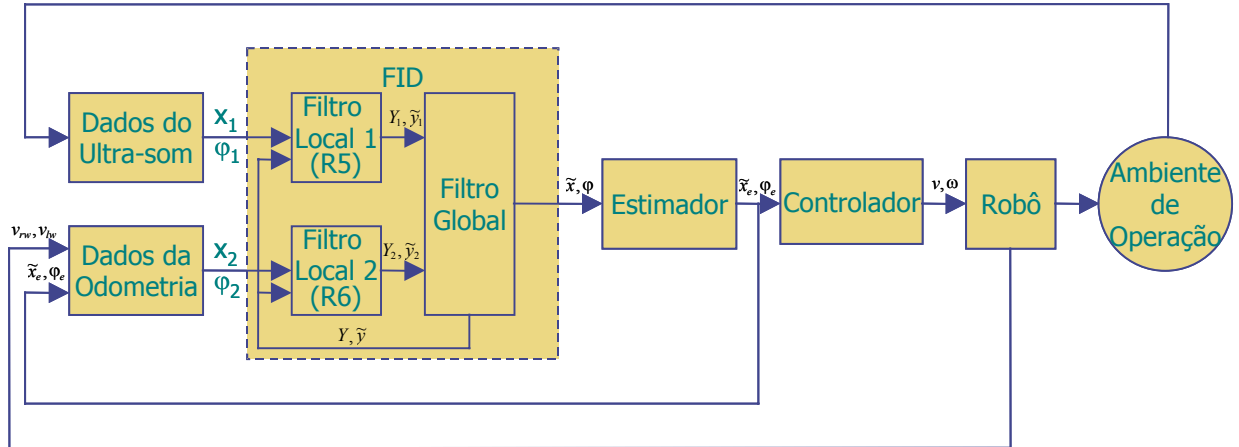


Figura 17 Processo de estimação das variáveis de estado  $\tilde{x}(t)$  e  $\varphi(t)$ .

A covariância associada a cada filtro local (Figura 17) é calculada utilizando lógica *fuzzy*, como descrito na Seção 3.2. São utilizadas duas variáveis *fuzzy* de entrada (anteriores): o produto das distâncias medidas à direita e à esquerda do robô ( $d_{dir} \times d_{esq}$ ), e *dif* (vide Figura 16).

Os consequentes são as covariâncias associadas aos filtros locais que estão recebendo como entradas os dados provenientes do sistema de sensoriamento ultra-sônico e os dados odométricos. Os consequentes são modelados como *singletons* [43], ou seja, como números reais.

A Figura 18 e a Figura 19 mostram as funções de pertinência das variáveis *fuzzy* de entrada e a Tabela 6 mostra as regras *fuzzy* que são utilizadas para determinar os consequentes (ou seja, as covariâncias associadas aos filtros locais), onde MP significa *muito pequeno*, P significa *pequeno*, G significa *grande* e MG significa *muito grande*. Na implementação foram considerados os valores MP = 1, P = 8, G = 27 e MG = 64.

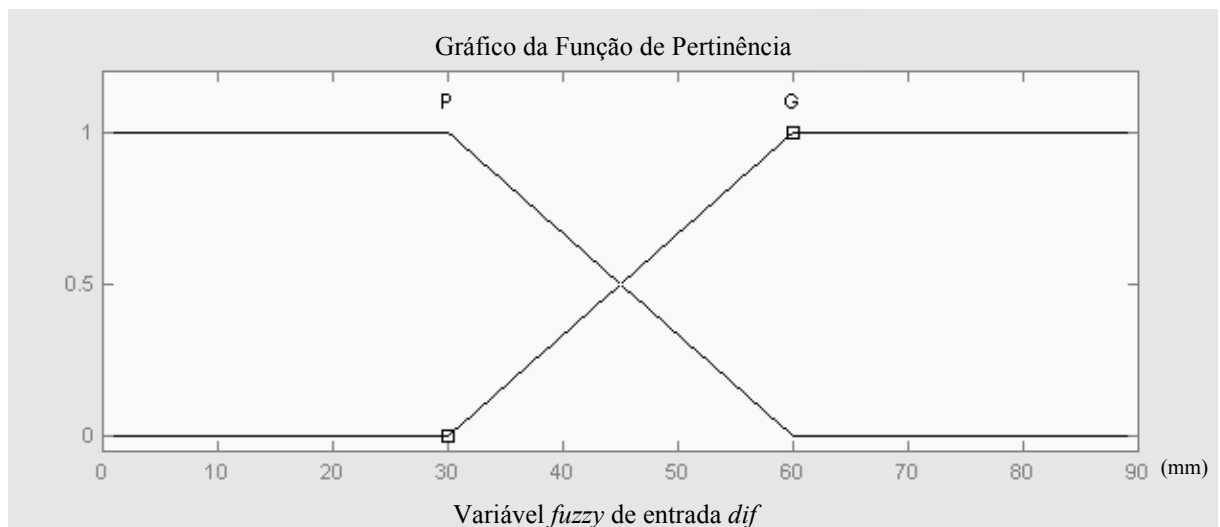


Figura 18 Função de pertinência da variável de *fuzzy* de entrada *dif*.

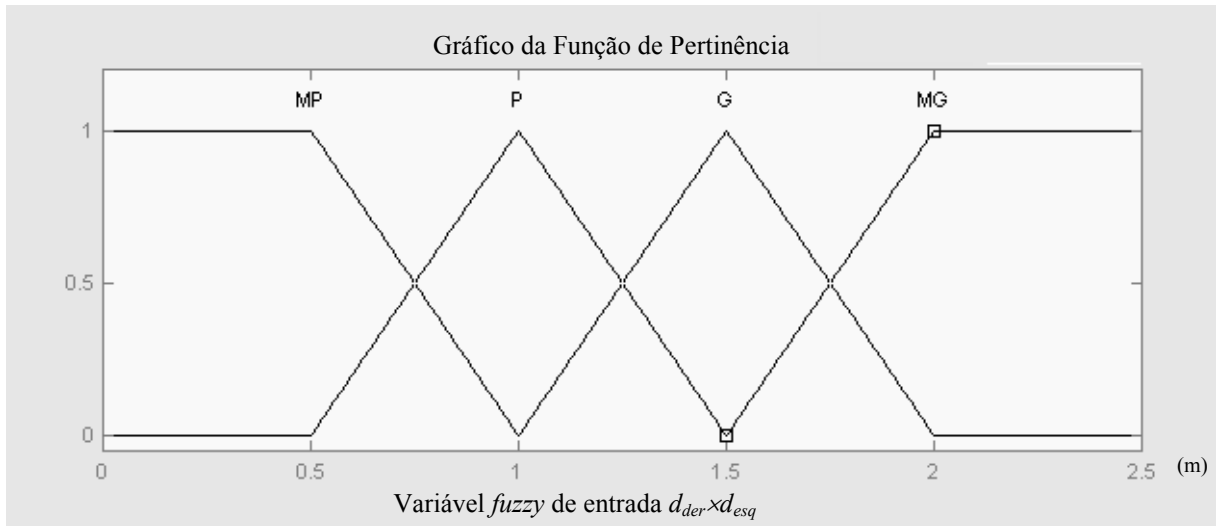


Figura 19 Função de pertinência da variável de *fuzzy* de entrada  $d_{dir} \times d_{esq}$ .

Tabela 6 Regras *fuzzy* utilizadas para determinar as covariâncias associadas aos filtros locais.

<i>Dif</i>	<i>P</i>	<i>G</i>
$d_{dir} \times d_{esq}$		
<b><i>MP</i></b>	R <sub>5</sub> =MP R <sub>6</sub> =MG	R <sub>5</sub> =G R <sub>6</sub> =P
<b><i>P</i></b>	R <sub>5</sub> =MP R <sub>6</sub> =MG	R <sub>5</sub> =MG R <sub>6</sub> =MP
<b><i>G</i></b>	R <sub>5</sub> =G R <sub>6</sub> =P	R <sub>5</sub> =MG R <sub>6</sub> =MP
<b><i>MG</i></b>	R <sub>5</sub> =MG R <sub>6</sub> =MP	R <sub>5</sub> =MG R <sub>6</sub> =MP

A *defuzzyficação* é realizada utilizando o método da média *fuzzy* [43], conforme (25).

Se o corredor é largo e o desvio angular do robô com relação ao eixo central do corredor é grande, a covariância associada aos dados provenientes dos sensores de ultra-som também deve ter um valor grande, de modo que o valor destes dados deve ter uma menor influência na saída do Filtro de Informação Descentralizado. Este caso corresponde às piores condições para as informações fornecidas pelo sistema de sensoriamento ultra-sônico. Sob estas condições os dados da odometria devem dominar a saída do Filtro de Informação Descentralizado, o que acontece através da atribuição de um valor pequeno para a covariância associada aos dados odométricos. Por outro lado, quando o corredor é estreito e o desvio angular do robô com relação ao eixo central do corredor é pequeno, os dados provenientes do sistema de sensoriamento ultra-sônico devem dominar a saída do Filtro de Informação Descentralizado. Para isso, a covariância associada aos dados provenientes dos sistema de sensoriamento ultra-sônico deve ser pequena e a covariância associada aos dados odométricos deve receber um valor elevado.

O estimador de estados mostrado na Figura 17 é implementado utilizando o Filtro de Informação Estendido [1], apresentado a seguir.

O Filtro de Informação estendido é o equivalente algébrico do Filtro de Kalman estendido [45], e foi introduzido pela primeira vez em [1]. Uma das principais vantagens do Filtro de Informação com relação ao Filtro de Kalman é o fato de que o Filtro de Informação é inicializado de uma forma mais simples (vide Seção 2.5). Isto é relevante sobretudo para sistemas não lineares, devido ao fato de que uma inicialização inadequada pode impedir a convergência do filtro [1].

Seja o modelo para o sistema

$$\begin{cases} \mathbf{x}(k) = \mathbf{F}(\mathbf{x}(k-1), (k-1)) + \mathbf{w}(k) \\ \mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \end{cases}$$

onde  $\mathbf{x}(k) = \begin{bmatrix} \tilde{x}(k) \\ \varphi(k) \end{bmatrix}$ , com ruídos  $\mathbf{w}(k)$  e  $\mathbf{v}(k)$  caracterizados por

$$E[\mathbf{w}(k)\mathbf{w}^T(k)] = \begin{cases} \mathbf{Q}(k) & i = k \\ 0 & i \neq k \end{cases}$$

$$E[\mathbf{v}(k)\mathbf{v}^T(k)] = \begin{cases} \mathbf{R}(k) & i = k \\ 0 & i \neq k \end{cases}$$

$$E[\mathbf{w}(k)\mathbf{v}^T(i)] = 0, \quad \forall k, i$$

e matrizes

$$\mathbf{F} = \begin{bmatrix} \omega(k) \\ u \sin[\varphi(k)] \end{bmatrix}$$

$$\mathbf{Q} = \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

O vetor de observação  $\mathbf{z}$  corresponde à saída do Filtro de Informação Descentralizado, como mostrado na Figura 17. As equações do Filtro de Informação estendido são [1]

Predição:

$$\hat{\mathbf{y}}(k | k-1) = \mathbf{Y}(k | k-1)\mathbf{F}(k, \hat{\mathbf{x}}(k-1 | k-1), (k-1))$$

$$\mathbf{Y}(k | k-1) = [\nabla \mathbf{F}_x(k)\mathbf{Y}^{-1}(k-1 | k-1)\nabla \mathbf{F}_x^T(k) + \mathbf{Q}(k)]^{-1}$$

Estimação:

$$\hat{\mathbf{y}}(k | k) = \hat{\mathbf{y}}(k | k-1) + \mathbf{i}(k)$$

$$\mathbf{Y}(k | k) = \mathbf{Y}(k | k-1) + \mathbf{I}(k)$$

onde  $\mathbf{I}(k) = \mathbf{R}^{-1}(k)$ ,  $\mathbf{i}(k) = \mathbf{R}^{-1}\mathbf{z}(k)$  e  $\nabla \mathbf{F}_x$  é o Jacobiano de  $\mathbf{F}$ .

As saídas do estimador de estados são as variáveis de estado  $\tilde{x}(t)$  e  $\varphi(t)$  que são passadas ao controlador de seguir corredores. As saídas do estimador de estados também são utilizadas para atualizar e corrigir as medições odométricas.

A seguir, são apresentados gráficos que demonstram os resultados do processo de estimação das variáveis de estado para o mesmo experimento detalhado na subseção seguinte, onde o robô navega ao longo de um corredor de 20 metros de comprimento e 1,40 metro de largura. A Figura 20 mostra a variável de estado  $\tilde{x}(t)$  calculada utilizando os dados odométricos corrigidos e utilizando as medições fornecidas pelo sistema de sensoriamento ultra-sônico. A Figura 21 mostra o sinal resultante do processo de fusão (saída do Filtro de Informação Descentralizado que tem por entradas os sinais mostrados na Figura 20), e a estimativa da variável  $\tilde{x}(t)$ . A Figura 22 mostra a variável de estado  $\varphi(t)$  calculada utilizando os dados odométricos corrigidos e utilizando as medições fornecidas pelo sistema de sensoriamento ultra-sônico. A Figura 23 mostra o sinal resultante do processo de fusão (saída do Filtro de Informação Descentralizado que tem por entradas os sinais mostrados na Figura 22), e a estimativa da variável  $\varphi(t)$ .

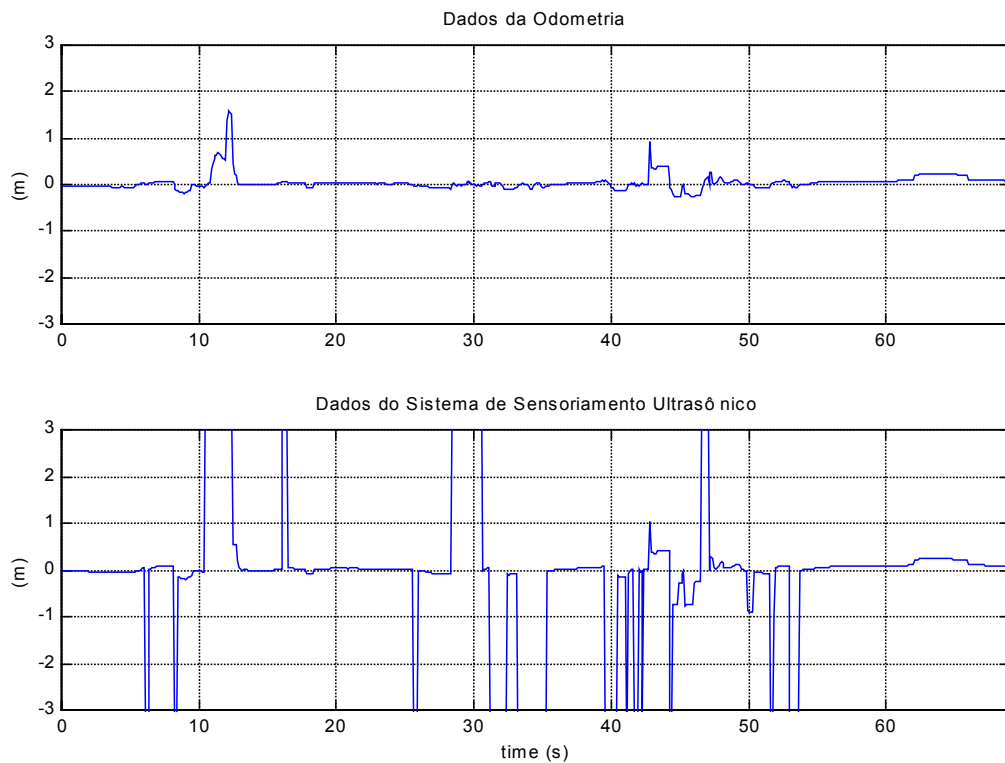


Figura 20 Variável de estado  $\tilde{x}(t)$  calculada utilizando os dados odométricos corrigidos (acima) e utilizando os dados fornecidos pelo sistema de sensoriamento ultra-sônico (abaixo).

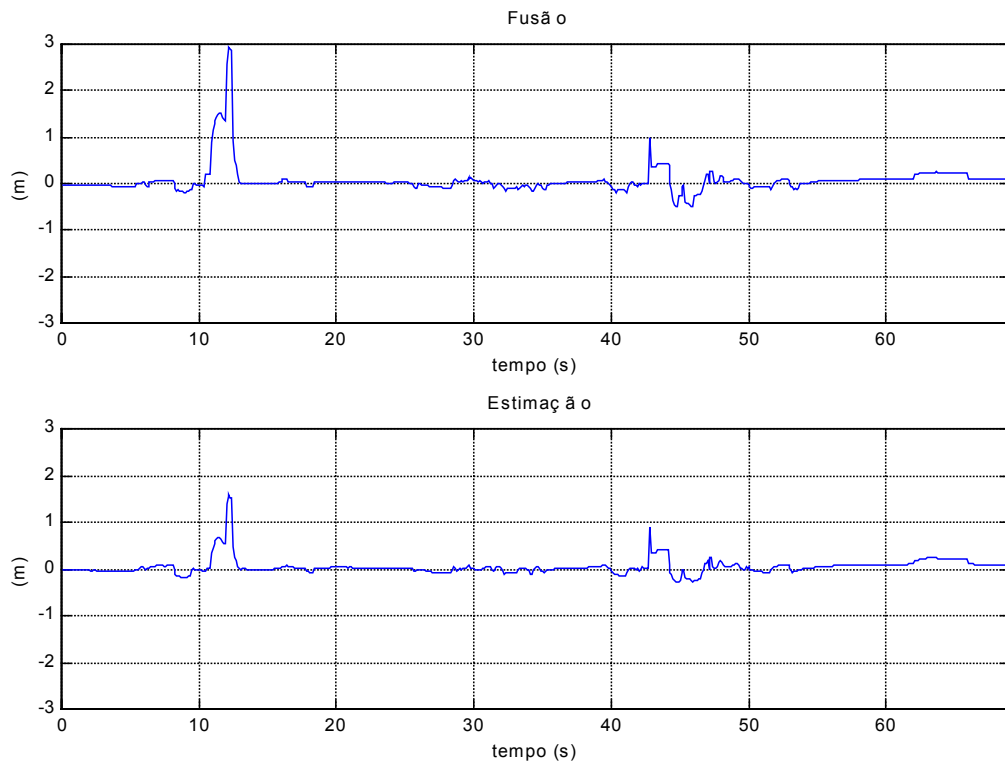


Figura 21 Sinal resultante do processo de fusão (acima), e a estimativa da variável  $\tilde{x}(t)$  (abaixo).

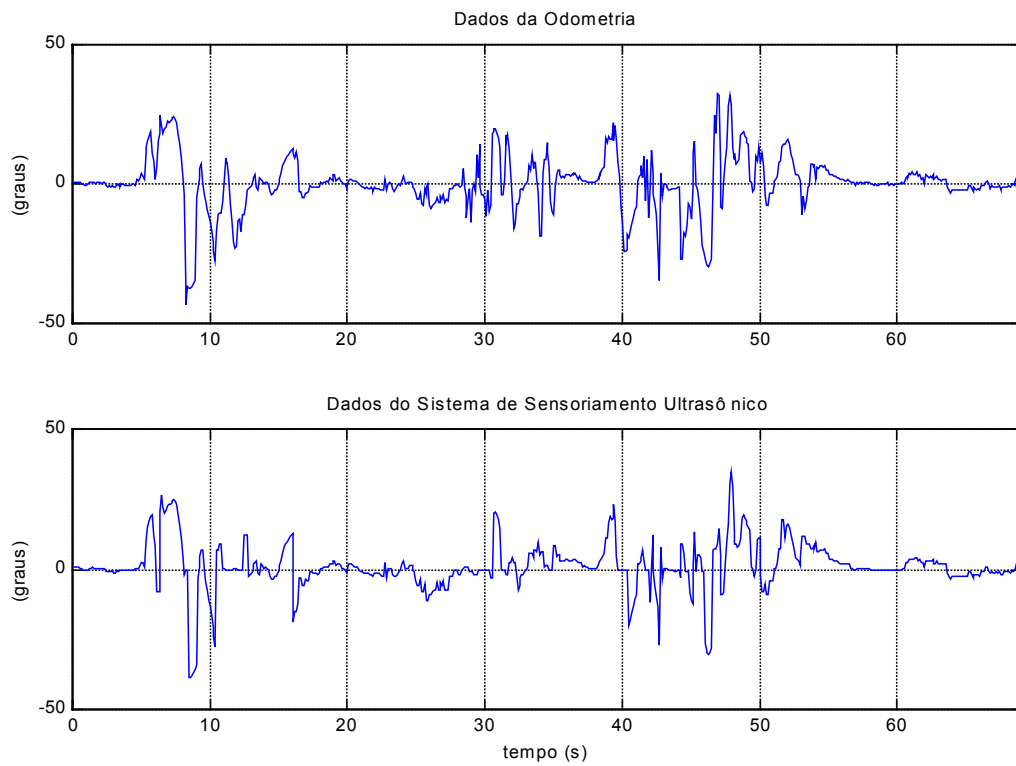


Figura 22 Variável de estado  $\varphi(t)$  calculada utilizando os dados odométricos corrigidos (acima) e utilizando os dados fornecidos pelo sistema de sensoriamento ultrassônico (abaixo).

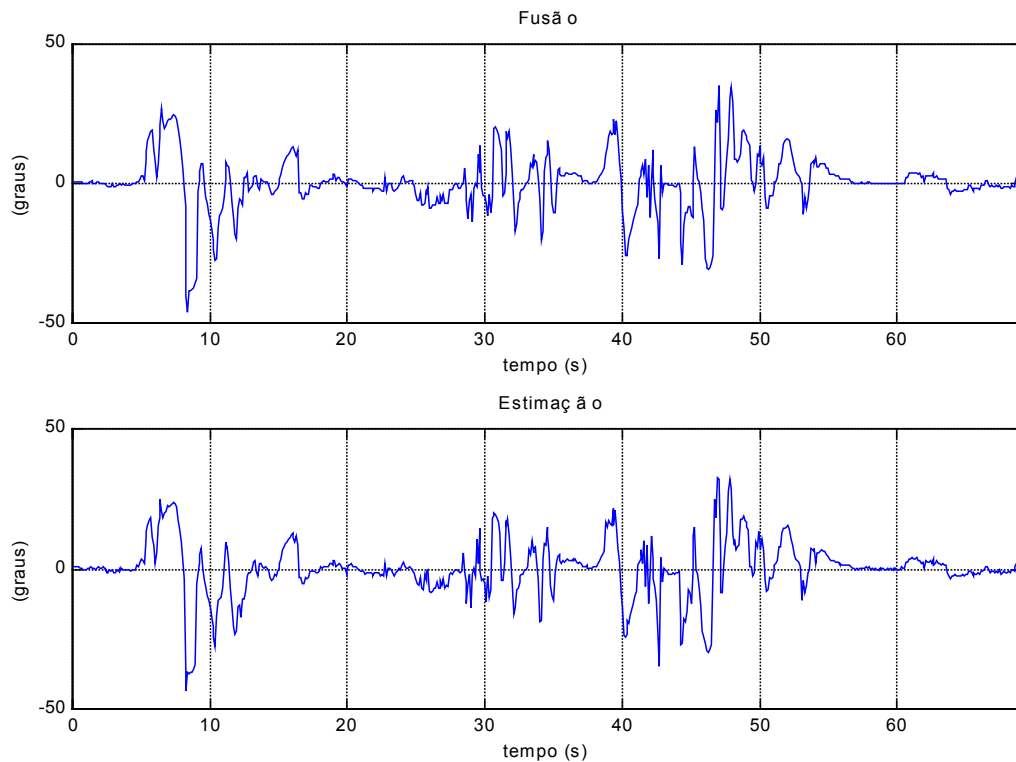


Figura 23 Sinal resultante do processo de fusão (acima), e a estimativa da variável  $\varphi(t)$  (abaixo).

### 5.1.5 Simulações e Resultados Experimentais

Simulações e experimentos foram realizados para comprovar os resultados teóricos e demonstrar o desempenho do sistema de controle projetado para desempenhar a tarefa de seguir corredores. Um resultado típico de uma simulação é mostrado na Figura 24, onde o controlador projetado é utilizado com as seguintes especificações e parâmetros de projeto:  $\omega_{\max}=60$  graus/s,  $a_1 = a_2 = 2$ ,  $k_1 = k_2 = 0.8$ ,  $u = 0.3$  m/s. As simulações demonstraram um bom desempenho do sistema de controle, com os erros de controle convergindo para zero, como mostra a Figura 24.

Os experimentos práticos foram realizados utilizando um robô móvel PIONEER 2DX (vide Seção 6.1). No experimento apresentado, o robô móvel PIONEER 2DX executa a tarefa de navegação ao longo de um dos corredores do *Instituto de Automática (INAUT) da Universidad Nacional de San Juan (UNSJ – Argentina)*, onde uma parte desta Tese de Doutorado foi desenvolvida. O corredor tem aproximadamente 1,40 m de largura e 20 m de comprimento, ao longo do qual existem aberturas laterais (junção com outros corredores e portas abertas).

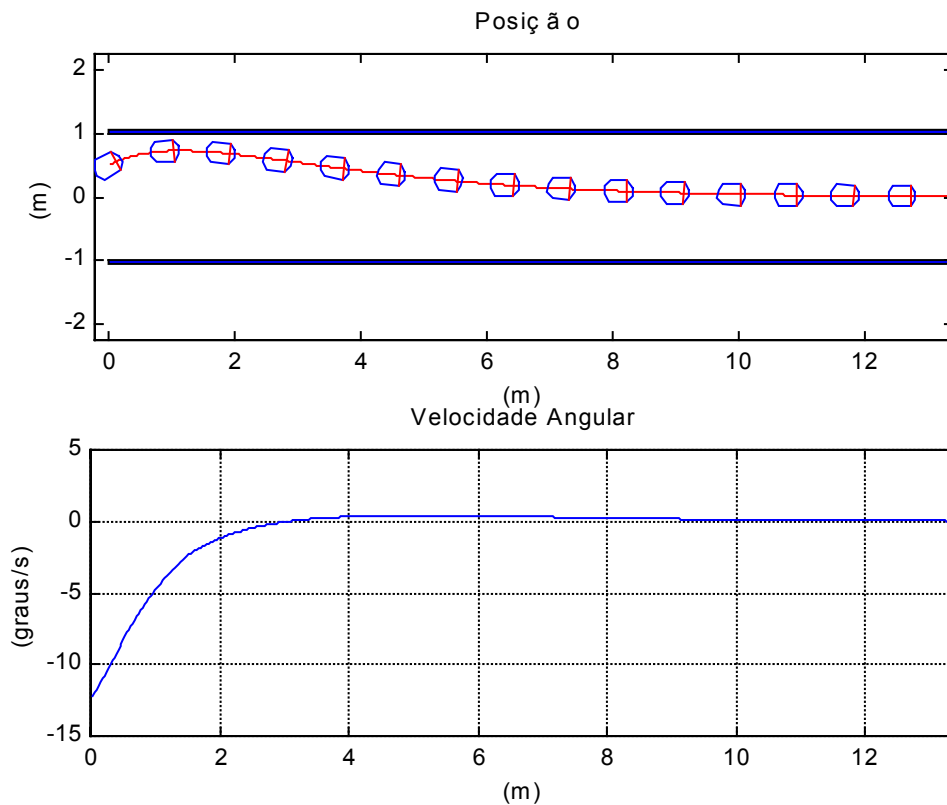


Figura 24 Simulação do controlador projetado para executar a tarefa de seguir corredores.

No corredor, foram colocados três obstáculos cilíndricos com aproximadamente 20 cm de diâmetro, os quais obstruem parcialmente a passagem do robô. O robô é capaz de evitar estes obstáculos, porque além do controlador utilizado para navegação em corredores, está sendo executado também o controlador utilizado para evitar obstáculos, que será descrito com maiores detalhes na seção 5.4. Neste experimento, as saídas dos dois controladores não estão sendo combinadas através de um processo de fusão, como no caso da nova arquitetura proposta no Capítulo 3. A saída do controlador de evitar obstáculos é incorporada ao sistema de controle como sendo uma perturbação, como mostra a Figura 25. Isto é possível porque o surgimento de obstáculos no caminho do robô em um ambiente previamente desconhecido pode ser considerado aleatório [23].

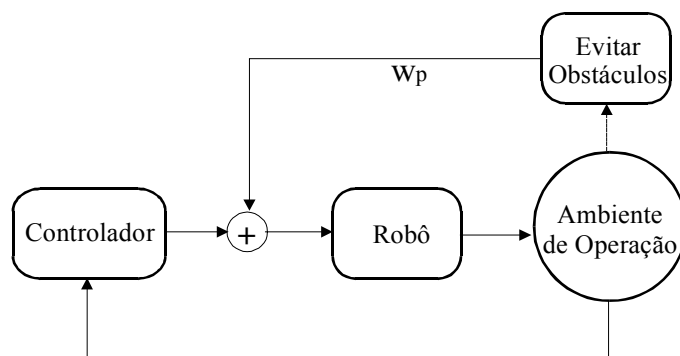


Figura 25 Estrutura utilizada para incorporar o controlador de evitar obstáculos como uma perturbação ao sistema de controle.

As seguintes especificações e parâmetros de projeto foram utilizados neste experimento:  $\omega_{\max}=60$  graus/s,  $a_1 = a_2 = 2$ ,  $k_1 = k_2 = 0.8$ ,  $u = 0.3\text{m/s}$ . A Figura 26 mostra o caminho percorrido pelo robô ao longo do corredor. Nela se pode também notar um conjunto de pontos que constituem uma reconstrução do ambiente de operação do robô móvel a partir dos dados coletados pelo seu sistema de sensoriamento ultra-sônico.

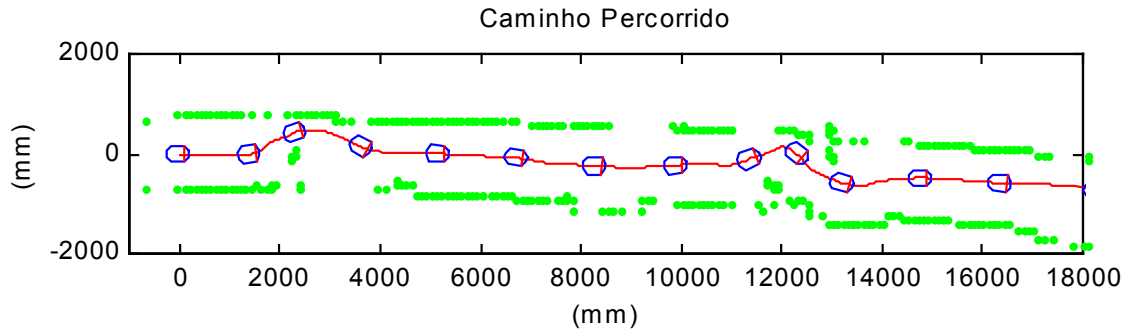


Figura 26 Caminho percorrido pelo robô em um dos experimentos realizados para demonstrar o desempenho do controlador projetado para seguir corredores, com a capacidade de evitar obstáculos (considerada como uma perturbação).

A Figura 27 mostra as variáveis de estado estimadas  $\tilde{x}(t)$  e  $\varphi(t)$  utilizadas pelo controlador. A Figura 28 representa os valores de velocidade angular produzidos pelo controlador (comandos de velocidade angular) para a tarefa de seguir corredores, e os valores de velocidade angular medidos pelo robô móvel.

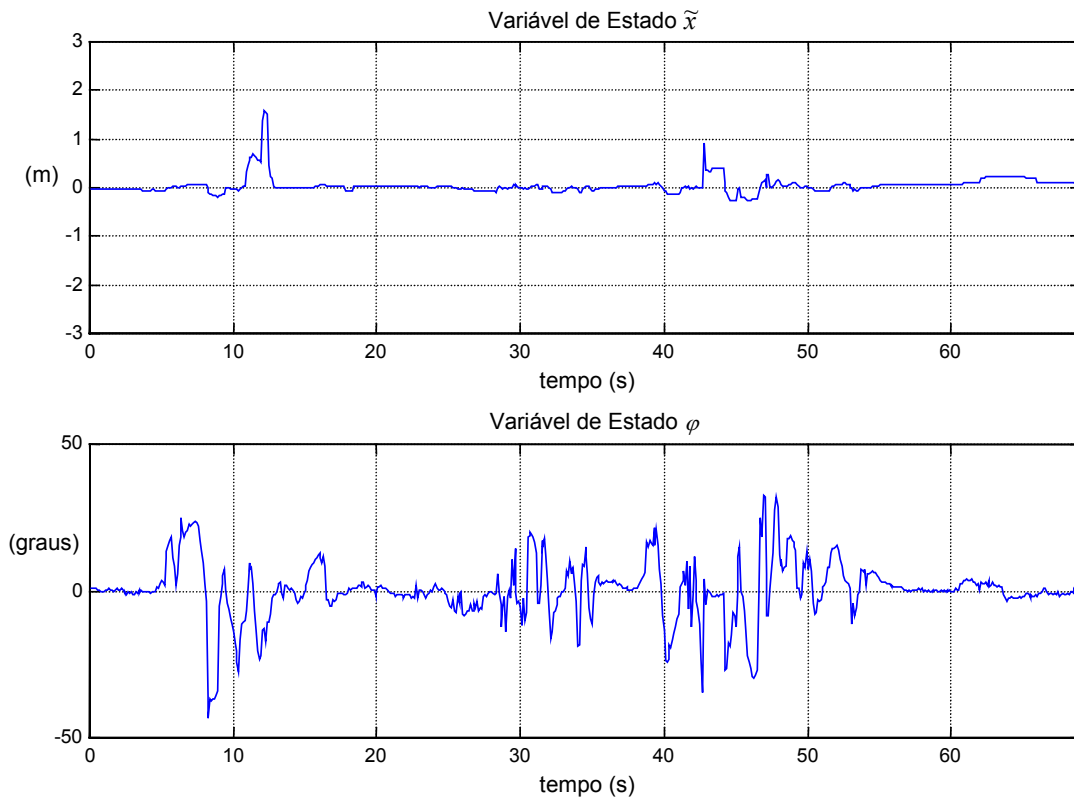


Figura 27 Variáveis de estado  $\tilde{x}(t)$  (acima) e  $\varphi(t)$  (abaixo) estimadas.



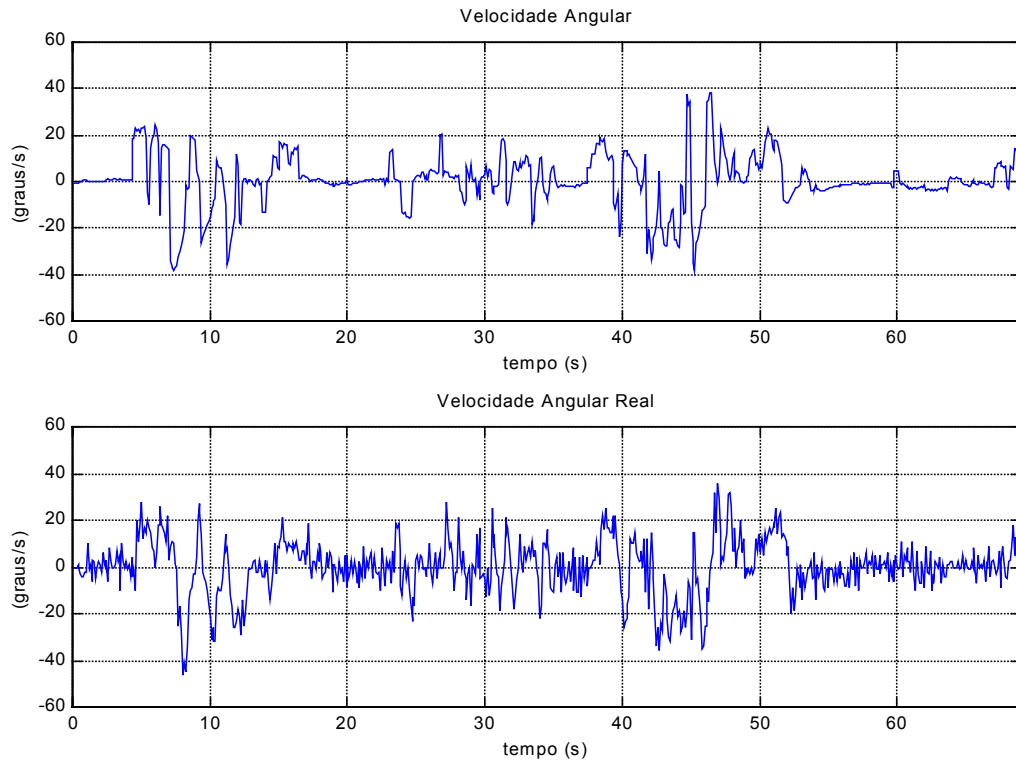


Figura 28 Velocidades angulares produzidas pelo controlador projetado para seguir corredores (acima) e velocidade angular medida pelo robô móvel (abaixo).

Como mostrado neste experimento e em vários outros realizados no INAUT, e que não foram apresentados aqui por questões de espaço, o robô apresentou um desempenho muito bom ao navegar por corredores reais, mesmo quando se deparava com obstáculos em seu caminho.

Este algoritmo de controle foi projetado para resultar em um sistema de controle estável no sentido de Lyapunov, o que é uma exigência para que ele possa ser incorporado a um sistema de controle projetado de acordo com a arquitetura de controle apresentada no Capítulo 3. A função energia associada a este controlador é dada por

$$V_{cor} = \frac{\varphi^2}{2} + |\tilde{x}| - a_1 \log(a_1 + |\tilde{x}|) + a_1 \log(a_1) \quad (47)$$

onde  $\tilde{x}(t)$  e  $\varphi(t)$  são as variáveis de estado e  $a_1$  é uma constante positiva. A equação (47) é obtida substituindo-se o resultado de (42) em (41) e desenvolvendo a equação resultante.

## 5.2 Controlador para Seguir Paredes

Tão importante quanto a habilidade de seguir corredores para um robô móvel é a capacidade de seguir paredes. Em [44] é desenvolvido um algoritmo de controle globalmente estável para o seguimento de paredes, baseado na informação disponibilizada por *encoders* incrementais e um sensor de ultra-som.

Se for considerado um conjunto de variáveis de estado diferente do que foi utilizado no projeto do controlador para navegar em corredores, desenvolvido na seção anterior, é possível adaptá-lo para dotar o robô da capacidade de seguir paredes. Neste caso, as variáveis de estado são definidas com relação à parede a ser seguida como  $\varphi$  e  $\tilde{d}$ , onde  $\varphi$  é o ângulo de orientação do robô com relação à parede (que é o mesmo ângulo já utilizado no caso da tarefa de navegar em corredores), e  $\tilde{d}$  representa a distância do robô a uma linha imaginária à distância desejada  $d_{par}$  da parede (vide Figura 29). A variável  $\tilde{d}$  pode ser calculada a partir dos dados fornecidos pelo sistema de sensoriamento ultra-sônico, conforme a equação

$$\tilde{d} = \pm \left\{ d_{par} - \min \left[ \left( \frac{y_{s0} + y_{s15}}{2} \right), \left( \frac{y_{s7} + y_{s8}}{2} \right) \right] \right\} \quad (48)$$

onde o sinal menos é considerado para o caso onde a parede a ser seguida está à direita do robô, e  $d_{par}$  é considerado igual a 50 cm nas simulações e experimentos.

As variáveis de estado para o controlador adaptado para seguir paredes,  $\varphi$  e  $\tilde{d}$ , são estimadas utilizando o mesmo processo de estimação utilizado para o caso da tarefa de navegação em corredores (vide Figura 17). Neste caso, os dados fornecidos pelo sistema de sensoriamento ultra-sônico são utilizados para calcular o ângulo  $\varphi$  utilizando (45) e (46), e a variável  $\tilde{d}$ , utilizando (48).

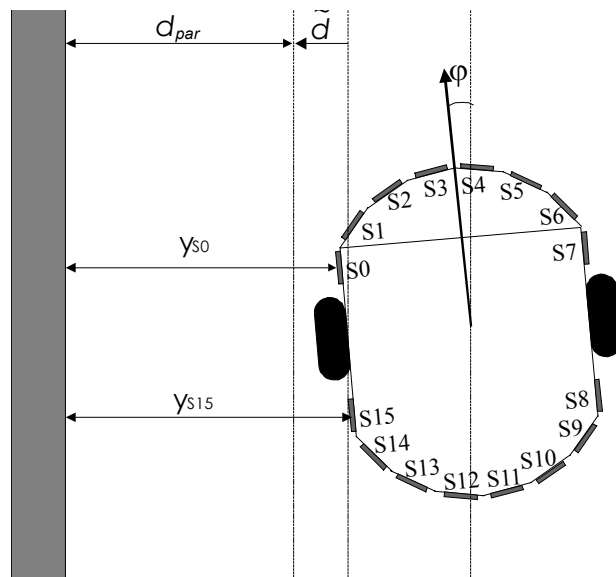


Figura 29 Variáveis de estado para o controlador de seguir paredes.

Para comprovar o desempenho do controlador adaptado para executar a tarefa de seguir paredes, é apresentado um experimento onde o robô inicia seguindo uma parede à sua esquerda. A distância desejada  $d_{par}$  é igual a 50 cm. Após alguns metros, o robô detecta uma parede à sua frente. O robô, então, evita a parede como se ela fosse um obstáculo (através do mesmo processo onde o controlador para evitar obstáculos é incorporado ao sistema como uma perturbação, conforme mostra a Figura 25), e se posiciona novamente a uma distância de 50 cm da parede que está seguindo.

As seguintes especificações e parâmetros de projeto foram utilizados neste experimento:  $\omega_{max}=60$  graus/s,  $a_1 = a_2 = 2$ ,  $k_1 = k_2 = 0.8$ ,  $u = 0.3$ m/s. A Figura 30 mostra o caminho percorrido pelo robô ao longo do corredor. Uma vez mais, o conjunto de pontos que aparece na Figura 30, constitui uma reconstrução do ambiente de operação do robô móvel a partir dos dados coletados pelo seu sistema de sensoriamento ultra-sônico.

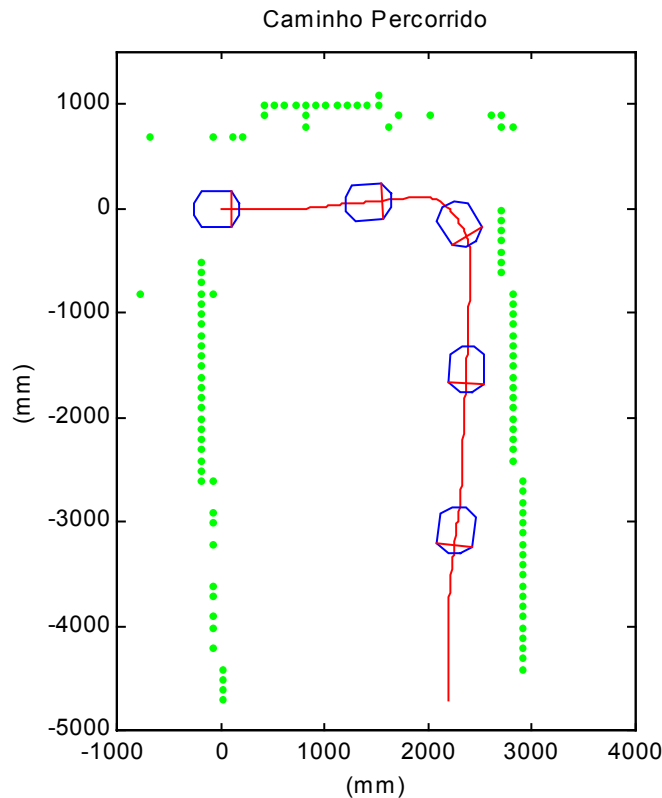


Figura 30 Caminho percorrido pelo robô.

A Figura 31 mostra as variáveis de estado  $\tilde{d}$  e  $\varphi(t)$  estimadas utilizadas pelo controlador. A Figura 32 apresenta os valores de velocidade angular produzidos pelo controlador (comandos de velocidade angular) para a tarefa de seguir paredes, e os valores de velocidade angular medidos pelo robô móvel.

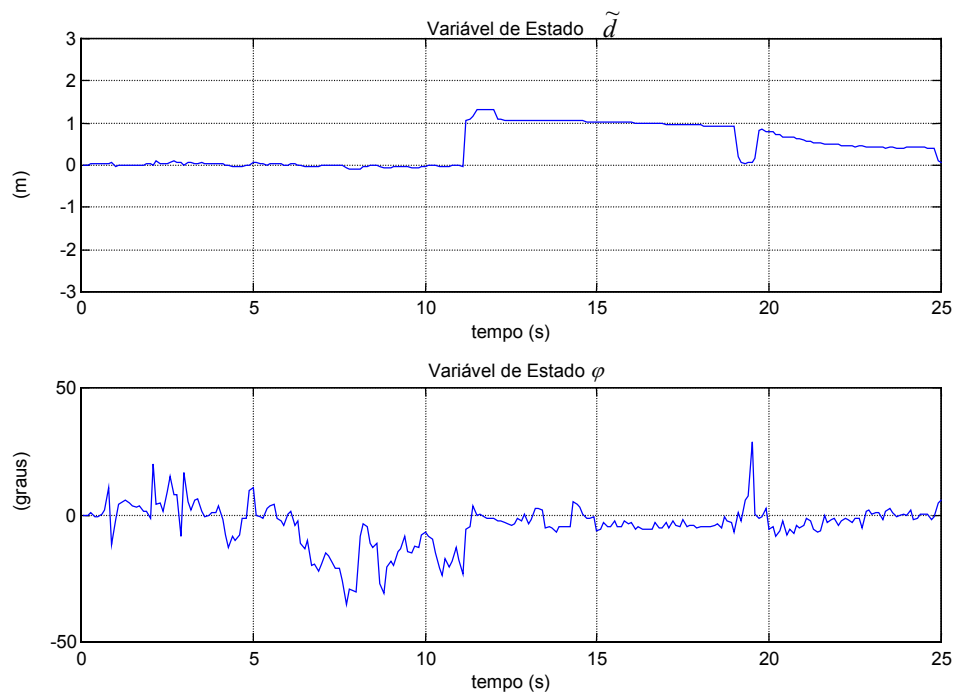


Figura 31 Variáveis de estado  $\tilde{d}$  (acima) e  $\varphi(t)$  (abaixo) estimadas.

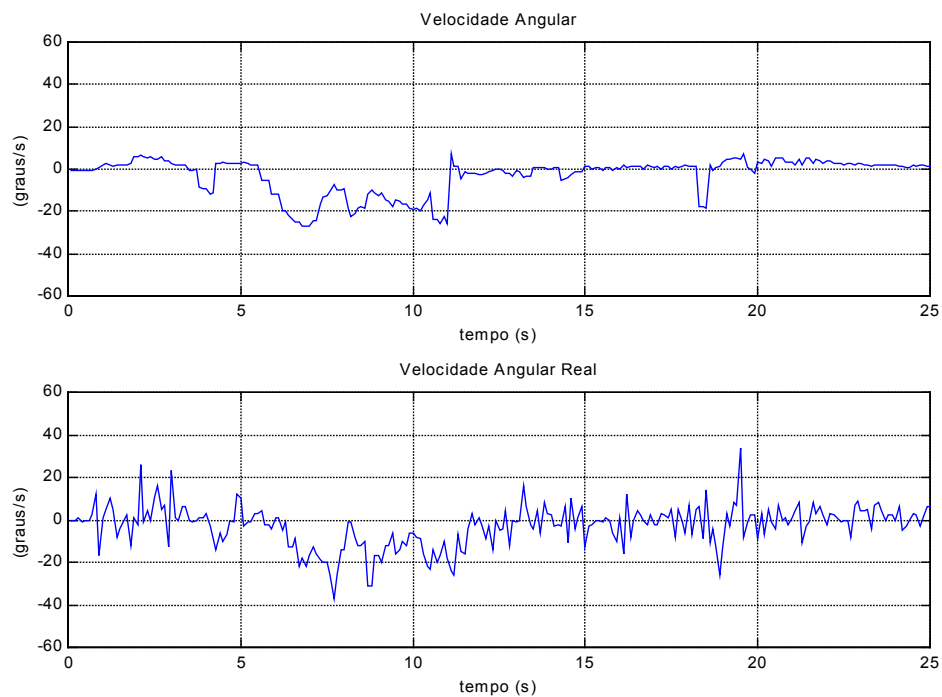


Figura 32 Velocidades angulares produzidas pelo controlador projetado para seguir paredes (acima) e velocidade angular medida pelo robô móvel (abaixo).

Como mostrado neste experimento e em vários outros realizados no INAUT, e que não foram apresentados aqui por questões de espaço, o robô apresentou um desempenho muito bom ao seguir paredes, mesmo quando se deparava com obstáculos em seu caminho.

Uma vez que este algoritmo de controle consiste em uma adaptação do algoritmo de controle desenvolvido para seguir corredores, e este por sua vez foi projetado para ser estável no sentido de Lyapunov, é correto afirmar que o controlador projetado para seguir paredes também conduz a um sistema de controle estável no sentido de Lyapunov. A função energia associada a este controlador é dada por

$$V_{par} = \frac{\varphi^2}{2} + |\tilde{d}| - a_1 \log(a_1 + |\tilde{d}|) + a_1 \log(a_1)$$

onde  $\tilde{d}(t)$  e  $\varphi(t)$  são as variáveis de estado e  $a_1$  é uma constante positiva.

### 5.3 Controlador de Posição Final

Este controlador é responsável por conduzir o robô de um ponto inicial até um ponto final desejado.

Seja o robô posicionado a uma distância diferente de zero do sistema de coordenadas de destino  $\langle g \rangle$ , conforme a Figura 14. Seu movimento em direção a  $\langle g \rangle$  é governado pela ação combinada tanto da velocidade angular  $\omega$  quanto do vetor velocidade linear  $u$ , que está sempre na direção de um dos eixos do sistema de coordenadas  $\langle a \rangle$  associado ao robô [24][48].

Assume-se que as variáveis de estado do sistema, (vide Figura 14), são:  $\rho$ , que é a distância entre o robô e a posição final desejada, e  $\alpha$ , que é o ângulo formado entre a direção de orientação do robô e a direção do ponto de destino. Logo, o sistema de equações que descreve o movimento do robô em termos destas duas variáveis de estado é dado por

$$\dot{\rho} = -u \cos \alpha$$

$$\dot{\alpha} = -\omega + \frac{u \sin \alpha}{\rho}$$

Os sinais de controle são dados por

$$u = k_u \rho \cos \alpha$$

$$\omega = \alpha + k_u \sin \alpha \cos \alpha$$

e as equações de malha fechada correspondentes são:

$$\dot{\rho} = -k_u \rho \cos^2 \alpha$$

$$\dot{\alpha} = -\alpha$$

$$\dot{\theta} = k_u \sin \alpha \cos \alpha$$

onde  $u$  é a velocidade linear,  $\omega$  é a velocidade angular e  $k_u$  é uma constante de proporcionalidade. Este controlador conduz a um sistema de controle assintoticamente estável, conforme prova em [24][48].

A função energia associada a tal controlador é

$$V_{pap} = \frac{\rho^2 + \alpha^2}{2}$$

onde  $\rho$  é a distância entre o robô e o ponto de destino e  $\alpha$  é o ângulo formado entre a direção de deslocamento do robô e o ponto de destino (vide Figura 14).

#### 5.4 Controlador para Evitar Obstáculos

O algoritmo desenvolvido para evitar obstáculos é baseado no controle baseado em impedância aplicado a robôs móveis [24][25][48]. Este controlador guia o robô de um ponto inicial até um ponto de destino localizado na origem do sistema de coordenadas  $\langle \mathbf{x}_d \rangle$ , (Figura 33). Ao detectar obstáculos presentes no ambiente de operação do robô, o controlador momentaneamente modifica a direção do ponto de destino para que o robô possa evitar os obstáculos detectados [24][25].

Com o objetivo de evitar os obstáculos, uma força de interação fictícia  $\mathbf{F}$  é gerada como sendo uma função da distância entre o robô e os obstáculos detectados [56], como mostrado na Figura 33. A força  $\mathbf{F}$  é composta por duas componentes: a componente perpendicular  $\mathbf{F}_r$  e a componente longitudinal à direção de deslocamento do robô  $\mathbf{F}_t$ .

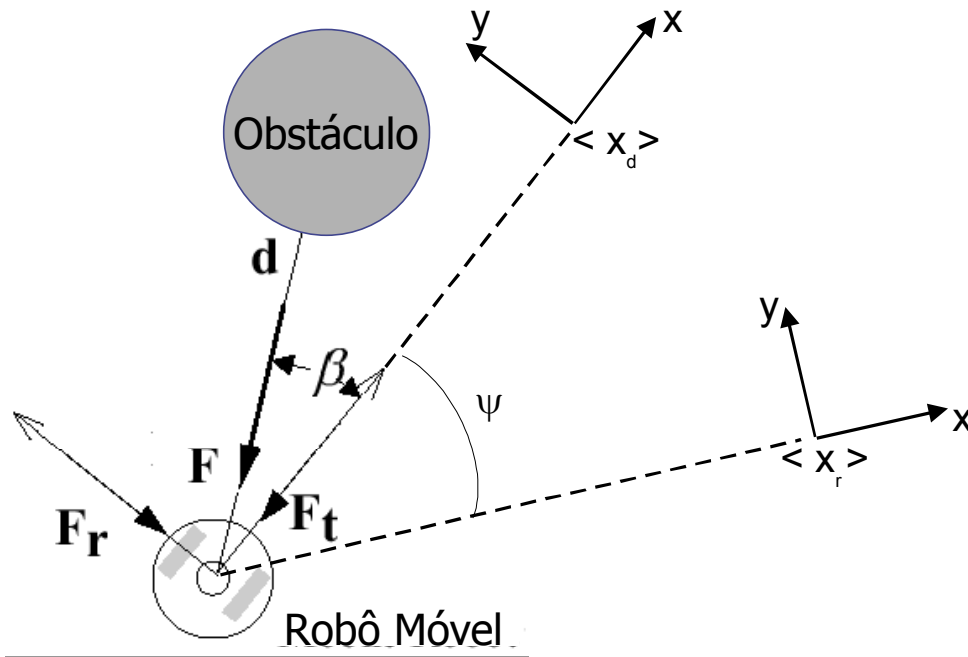


Figura 33 A força fictícia  $\mathbf{F}$ .

A magnitude da força  $\mathbf{F}$  é calculada como [26]

$$F = a - b \cdot [d - d_{\min}]^2$$

onde  $a$  e  $b$  são constantes positivas, tais que  $a - b \cdot [d_{\max} - d_{\min}]^2 = 0$ , o valor  $d_{\max}$  é a distância máxima entre o robô e um obstáculo detectado que deverá produzir uma força repulsiva,  $d_{\min}$  é a mínima distância que o sistema de sensoriamento é capaz de medir, e  $d$  é a distância medida entre o robô e o obstáculo detectado. Os valores utilizados para  $d_{\max}$  e  $d_{\min}$  foram, respectivamente, 100 e 20 cm.

A impedância correspondente é definida como [24][48]

$$Z(s) = Bs + K$$

$$x_a = Z(s)^{-1} \cdot \mathbf{F}_t$$

onde  $B$  e  $K$  são constantes positivas. A constante  $B$  representa a amortização, e  $K$  a elasticidade na interação entre o robô móvel e o obstáculo detectado. Uma variável auxiliar  $x_a$  é então utilizada para gerar o ângulo de rotação  $\psi$  do ponto de destino com relação ao centro do robô, ou seja

$$\psi = x_a \cdot \text{sign}(\mathbf{F}_r)$$

Então, a transformação

$$\mathbf{x}_r = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_d$$

é aplicada para calcular o novo ponto de destino  $\mathbf{x}_r$ . Um novo erro de posição é definido, agora com relação ao novo ponto de destino, e é igual a  $\tilde{\mathbf{x}}_n = \mathbf{x}_r - \mathbf{x}_c$ , sendo que  $\mathbf{x}_c^T = [x_c \ y_c \ \varphi]$  é o vetor de coordenadas cartesianas do robô. Quando a força fictícia é igual a zero  $\mathbf{x}_r = \mathbf{x}_d$ , e o objetivo do laço de controle de movimento é atingido, significando que  $\tilde{\mathbf{x}} = (\mathbf{x}_d - \mathbf{x}_c) \rightarrow 0$  quando  $t \rightarrow \infty$ . A Figura 34 representa o diagrama de blocos do sistema de controle baseado em impedância.

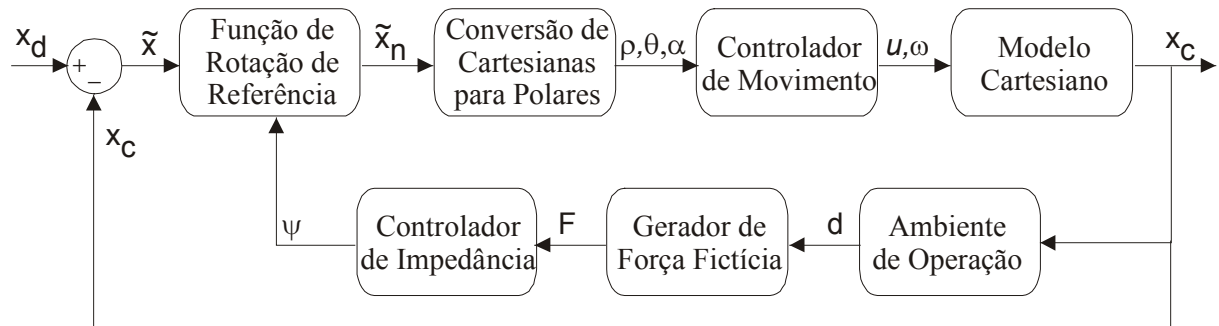


Figura 34 Diagrama de blocos do sistema de controle baseado em impedância [24][48].

Em resumo, quando um ou mais obstáculos são detectados, uma força fictícia é calculada (inversamente proporcional à distância entre o robô e o obstáculo) para cada obstáculo detectado, e uma força resultante é calculada através de soma vetorial. De acordo com essa força fictícia, o ponto de destino do robô é rotacionado, permitindo evitar os obstáculos detectados.

O referido algoritmo é adequado para guiar um robô de um ponto inicial até um ponto de destino, enquanto evita eventuais obstáculos [24][25][48]. Para adaptar este algoritmo para executar somente a tarefa de evitar obstáculos, basta executar uma simples mudança, que consiste em posicionar o ponto de destino sempre a uma distância de um metro do robô móvel, e sempre na direção para a qual o robô estiver orientado em cada momento. Desse modo, na ausência de obstáculos este algoritmo de controle não terá nenhuma influência na navegação do robô. Porém, quando for detectado um ou mais obstáculos, o controlador de impedância irá rotacionar temporariamente o ponto de destino, permitindo ao robô evitar os referidos obstáculos.

Conforme mostrado em [24][25][48], este algoritmo de controle é estável no sentido de Lyapunov. A função energia associada a este controlador, por sua vez, é


$$V_{obs} = \frac{\alpha^2}{2}$$

onde  $\alpha$  é o ângulo formado entre a direção de deslocamento do robô e o ponto de destino (vide Figura 14).



## Capítulo 6

# Simulações e Resultados Experimentais



Para verificar o desempenho da arquitetura de controle proposta e verificar se a conjectura de estabilidade proposta na Seção 4.2 se cumpre, foi escrito um programa em linguagem C para implementar a arquitetura de controle apresentada na Figura 9 e foram executados vários testes nos quais o robô deveria chegar em um ponto de destino em ambientes de operação com quatro configurações diferentes. Os testes foram realizados por simulação computacional e também utilizando um robô móvel PIONEER 2DX.

### 6.1 O Robô Móvel PIONEER 2DX

Os experimentos foram executados usando um robô móvel PIONEER 2DX ([www.activmedia.com](http://www.activmedia.com)), que possui 16 sensores de ultra-som (dos quais somente 10 são efetivamente usados) e uma câmera CCD (não utilizada nos experimentos). As rodas motorizadas e os sensores de ultra-som são controlados por um microcontrolador 88C166 de 20 MHz, fabricado pela Siemens. A navegação é controlada por um computador de bordo (um K6-II de 500 MHz) executando o sistema de controle mostrado na Figura 9. A Figura 35 mostra o robô móvel PIONEER 2DX.

Para as simulações, foi utilizado um simulador que acompanha o robô móvel PIONEER 2DX, denominado *Saphira*. O simulador busca levar em consideração modelos de erro associados aos sensores de ultra-som e à odometria do robô. O modelo dinâmico do robô porém, aparentemente não é levado em consideração pelo simulador, pois os valores de velocidades lineares e angulares medidos pelo robô são muito distintos dos valores obtidos na simulação.



Figura 35 Robô móvel PIONEER 2DX.

## 6.2 Simulações e Experimentos Realizados

Foram executados vários testes, os quais consistiam tanto em simulações, como em experimentos práticos, e alguns deles são apresentados neste capítulo. Tanto nas simulações como nos experimentos, o tempo de duração do laço de controle implementado é de 100 ms.

Para cada teste realizado, inicialmente é apresentado um desenho ilustrando a tarefa que o robô deve executar. Em seguida, apresenta-se uma tabela com os índices utilizados para avaliar o desempenho do robô [39]. A tabela mostra os valores dos índices resultantes tanto da simulação quanto do experimento, incluindo o valor ideal de cada índice. O índice *segurança* indica a menor distância medida pelos 16 sensores de ultra-som ao longo de todo o trajeto, indicando assim o risco de colisão. O índice *velocidade média* indica a velocidade linear média do robô ao longo do trajeto. Finalmente, o índice *suavidade* é medido calculando o valor médio do módulo da diferença entre o valor atual e anterior da orientação do robô, indicando o grau de suavidade com que as manobras são executadas. A tabela também apresenta a distância percorrida e a duração da simulação e do experimento. Após a tabela, são apresentadas onze figuras para a simulação e outras onze (equivalentes) para o experimento. A primeira delas mostra as covariâncias associadas a cada um dos quatro controladores envolvidos no processo de fusão. A segunda figura apresenta os mesmos gráficos das covariâncias, porém sobrepostos em um único gráfico. Na terceira figura, são apresentadas as velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô. A quarta figura apresenta a velocidade linear resultante do processo de fusão e a velocidade linear medida pelo robô, sobrepostas em um único gráfico. Na quinta figura são apresentadas as velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô. A sexta figura apresenta a velocidade angular resultante do processo de fusão e a velocidade angular medida pelo robô, sobrepostas em um único gráfico. A sétima figura apresenta a função energia

associada a cada controlador. Na oitava figura apresenta-se a função energia do sistema e a indicação das alterações das fases de navegação<sup>10</sup>. A nona figura apresenta somente a função energia do sistema. Na décima apresenta-se como evolui ao longo do tempo o estado de cada controlador (ativo – valor ‘1’, ou inativo – valor ‘0’). Finalmente, na décima primeira figura apresenta-se o percurso realizado pelo robô.

### 6.2.1 Teste #1

Neste teste, o robô é posicionado em um corredor com três metros de largura e deve chegar a um ponto localizado cinco metros diretamente à sua frente. No entanto, no meio do trajeto entre o ponto de partida e o ponto de destino, é posicionado um obstáculo cilíndrico com vinte centímetros de diâmetro. É importante notar que, neste caso, o controlador de seguir corredores não predomina porque considera-se que um corredor de três metros de largura é muito largo (conforme as regras *fuzzy* apresentadas na Tabela 4 e na Tabela 5), e, portanto, é mais importante buscar o ponto final do que seguir o corredor.



Figura 36 Teste #1.

Tabela 7 Índices de Avaliação de Desempenho do Teste #1.

Índice	Simulação	Experimento	Valores Ideais
Segurança	260 mm	252 mm	1250 mm
Velocidade Média	222 mm/s	224 mm/s	300 mm/s
Suavidade	0,82°	0,90°	0°
Distância Percorrida	5,04 m	4,77 m	5,00 m
Tempo	22,70 s	21,30 s	16,67 s

<sup>10</sup> As alterações das fases de navegação são indicadas através de linhas verticais. Quando ocorre uma mudança de fase de navegação, ocorre a subida de uma linha vertical. Esta permanece com um valor diferente de zero enquanto ocorrerem mudanças nas fases de navegação em intervalos de tempo sucessivos. Quando não ocorrer uma mudança de fase de navegação, a linha vertical mencionada anteriormente não é representada.

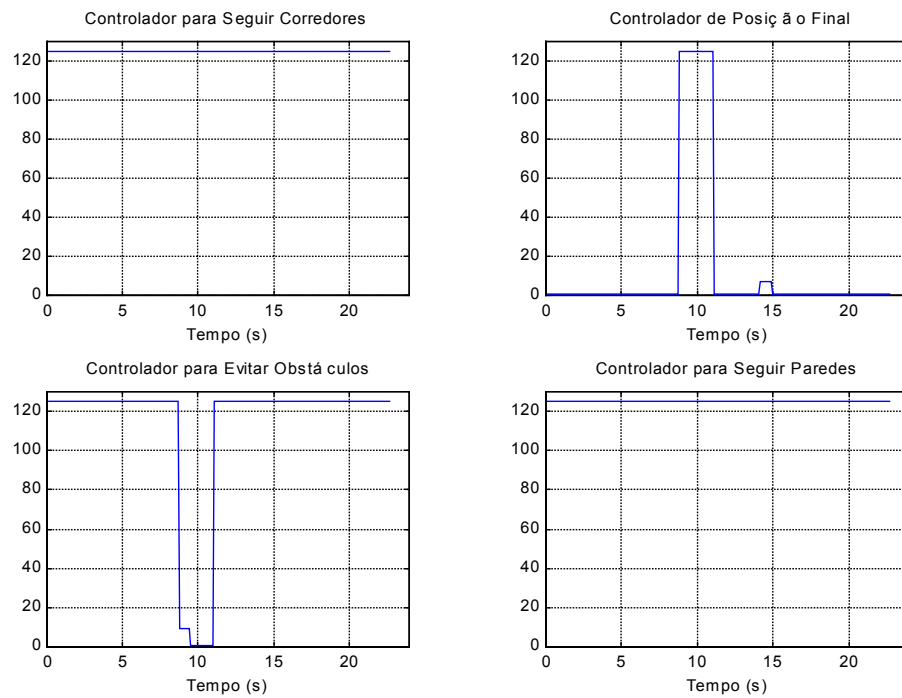


Figura 37 Covariâncias associadas aos controladores (Simulação #1).

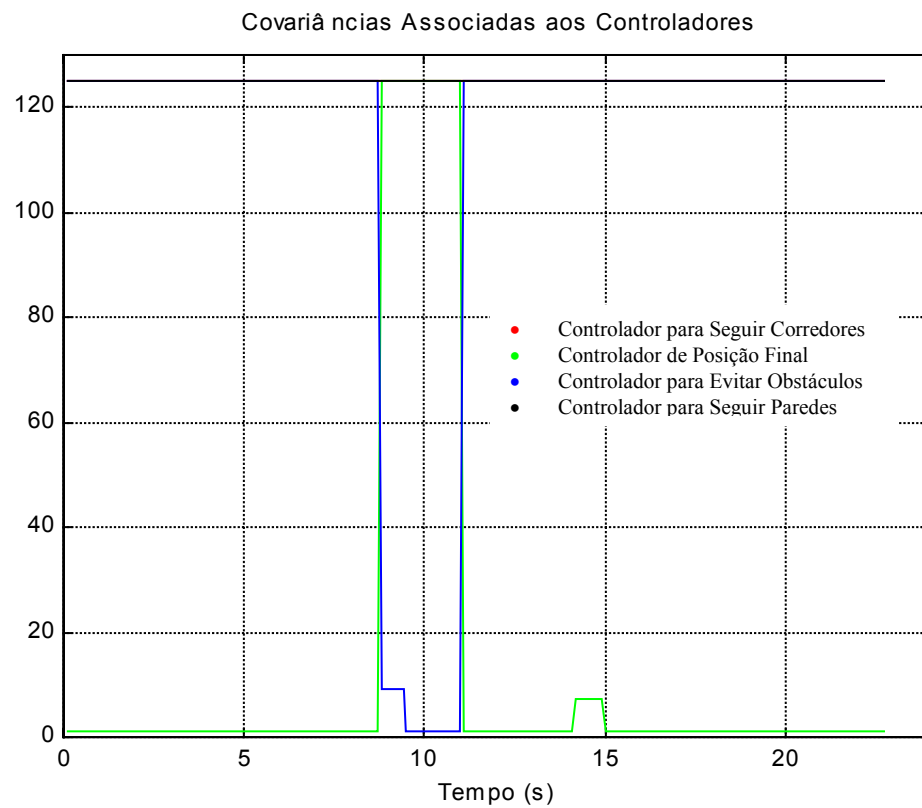


Figura 38 Covariâncias associadas aos controladores sobrepostas (Simulação #1)<sup>11</sup>.

<sup>11</sup> Neste experimento a covariância associada aos controladores de seguir paredes e seguir corredores é igual ao valor máximo ao longo de todo o tempo (vide Figura 37).

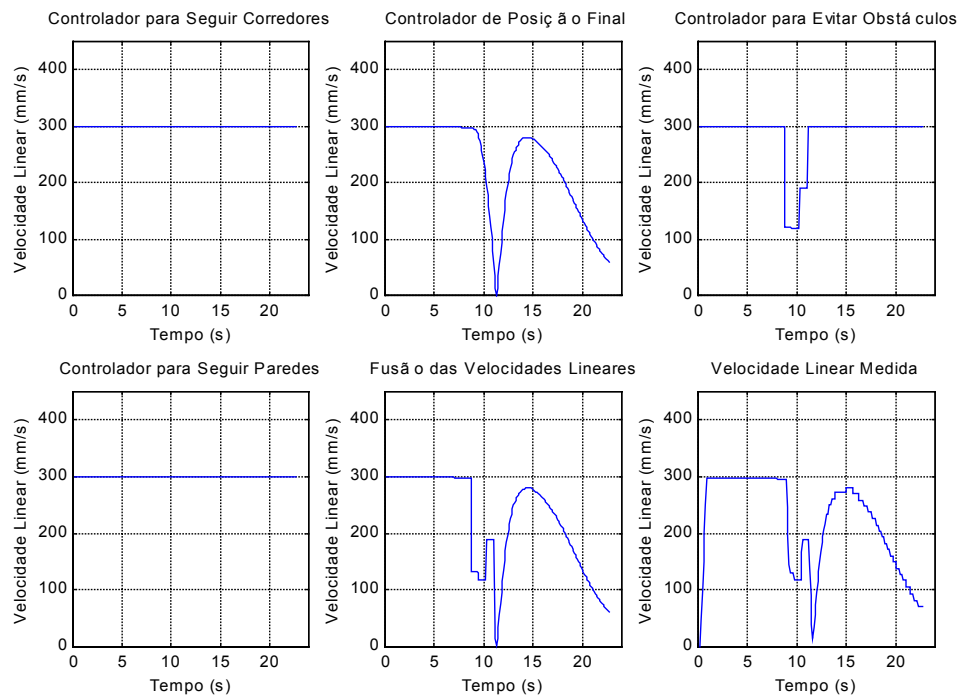


Figura 39 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #1).

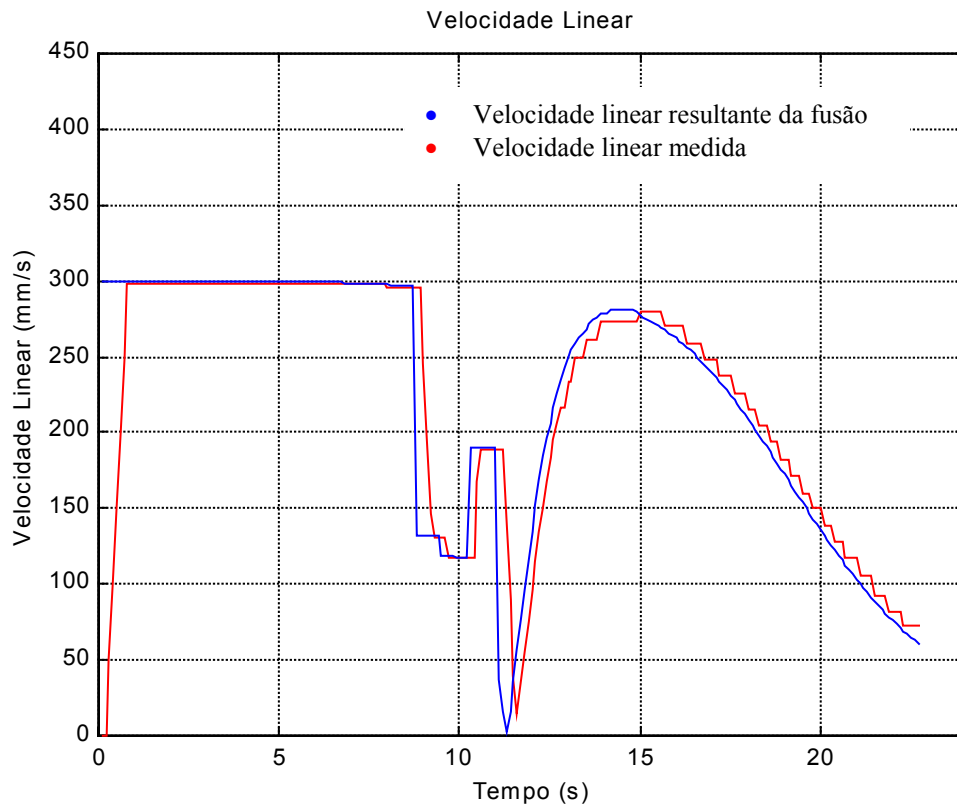


Figura 40 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #1).

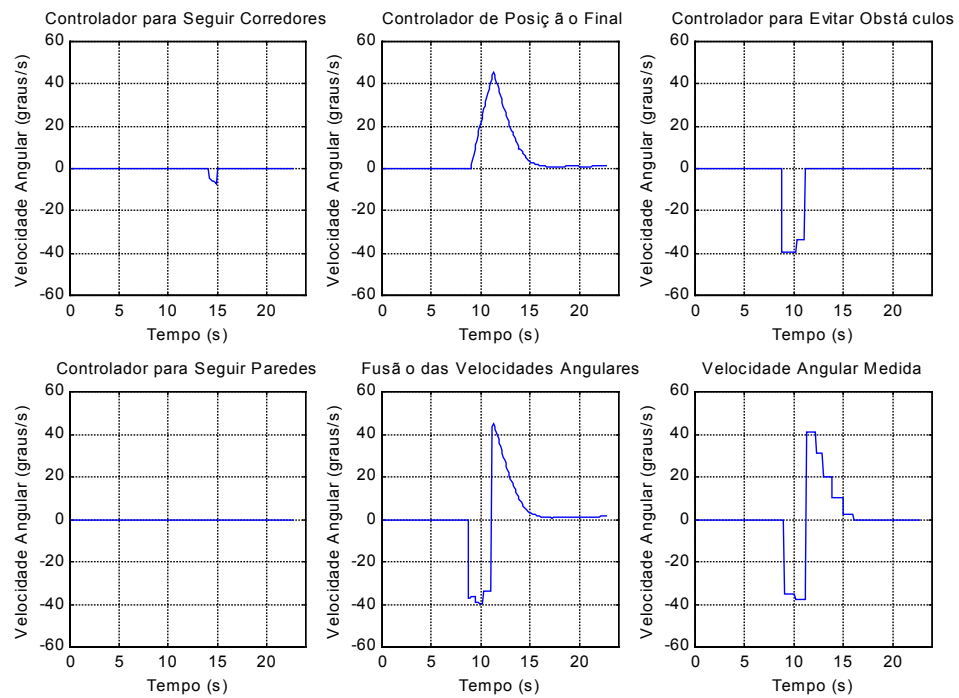


Figura 41 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #1).

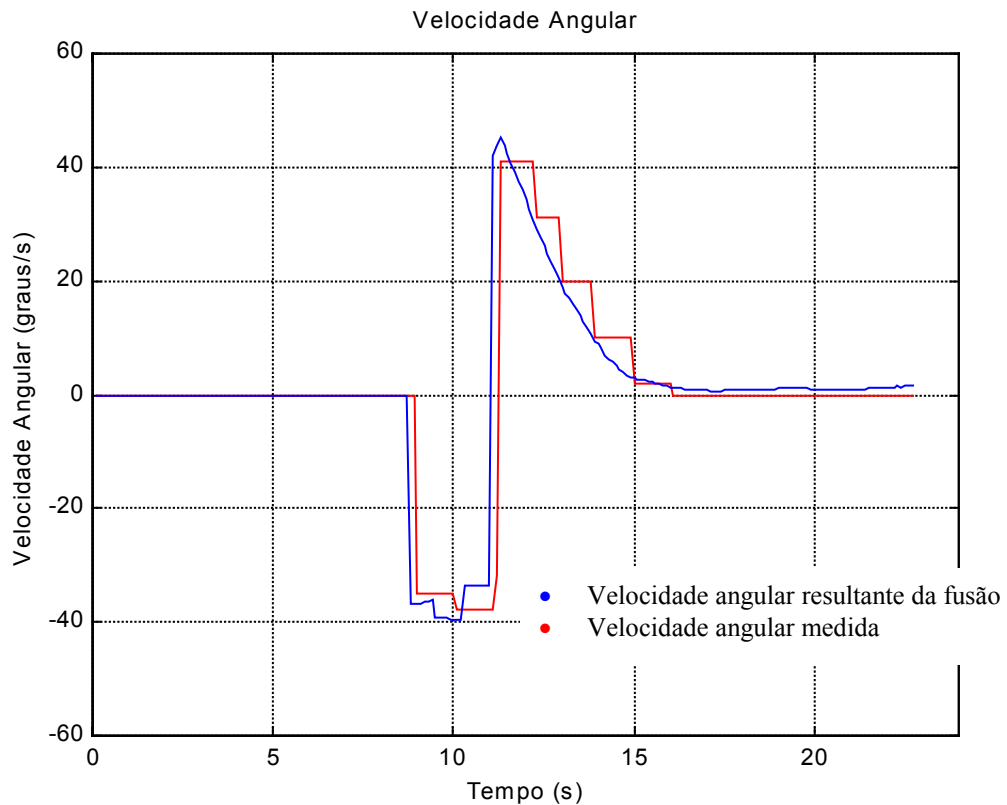


Figura 42 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Simulação #1).

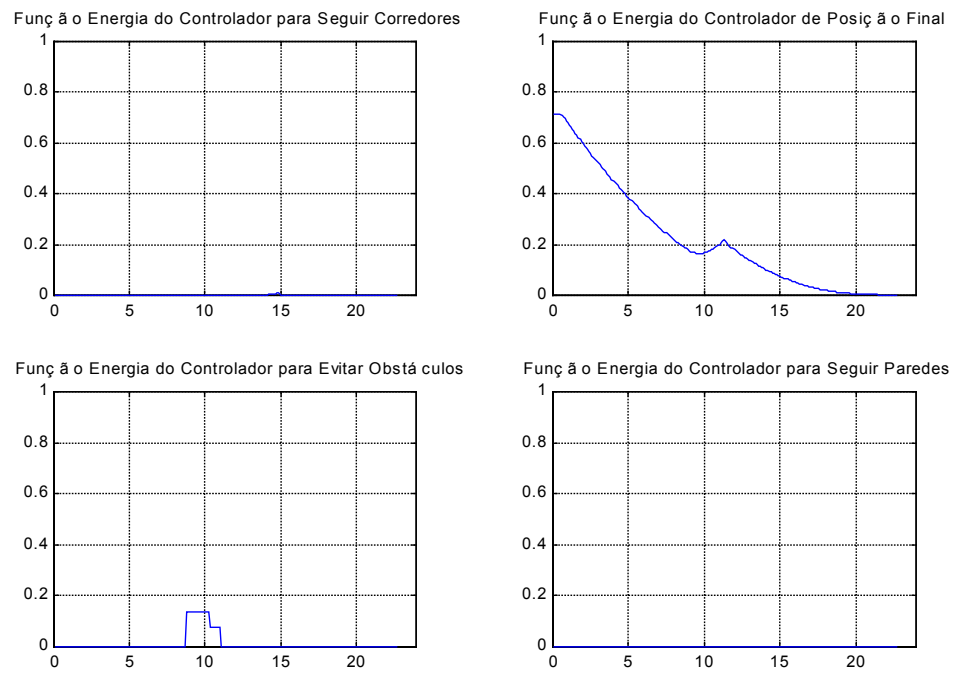


Figura 43 Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #1).

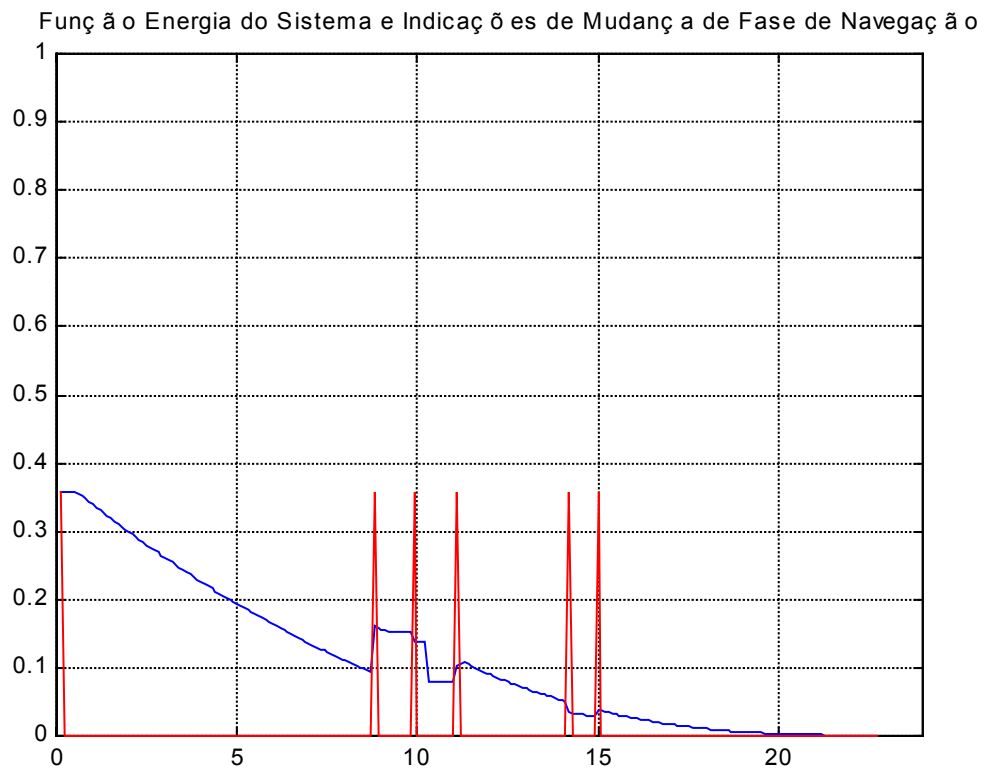


Figura 44 Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #1).

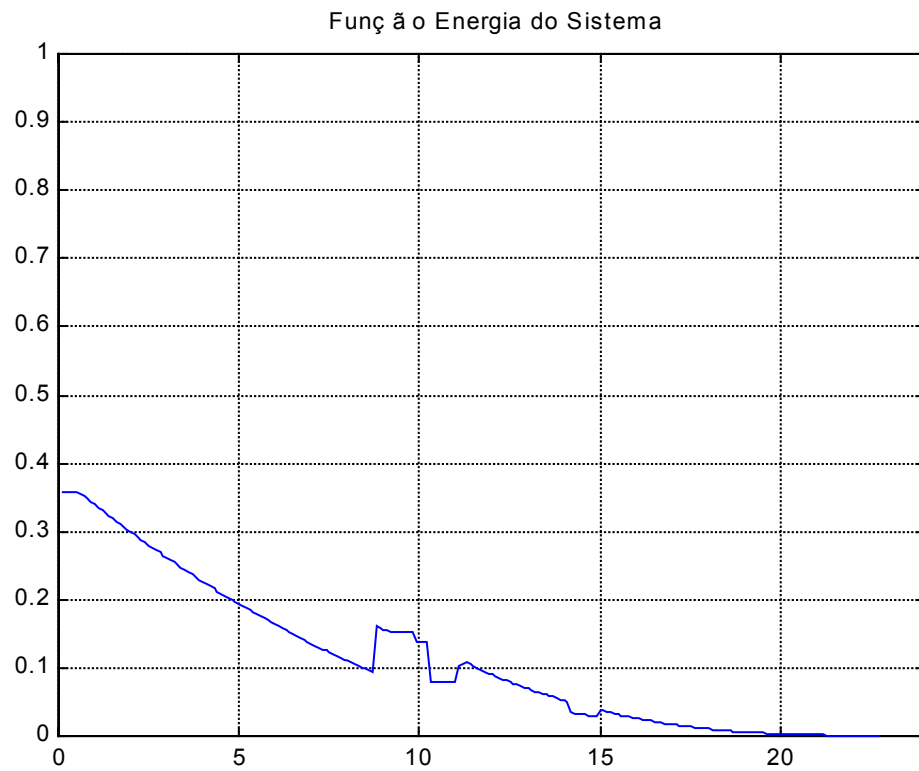


Figura 45 Função energia do sistema (Simulação #1).

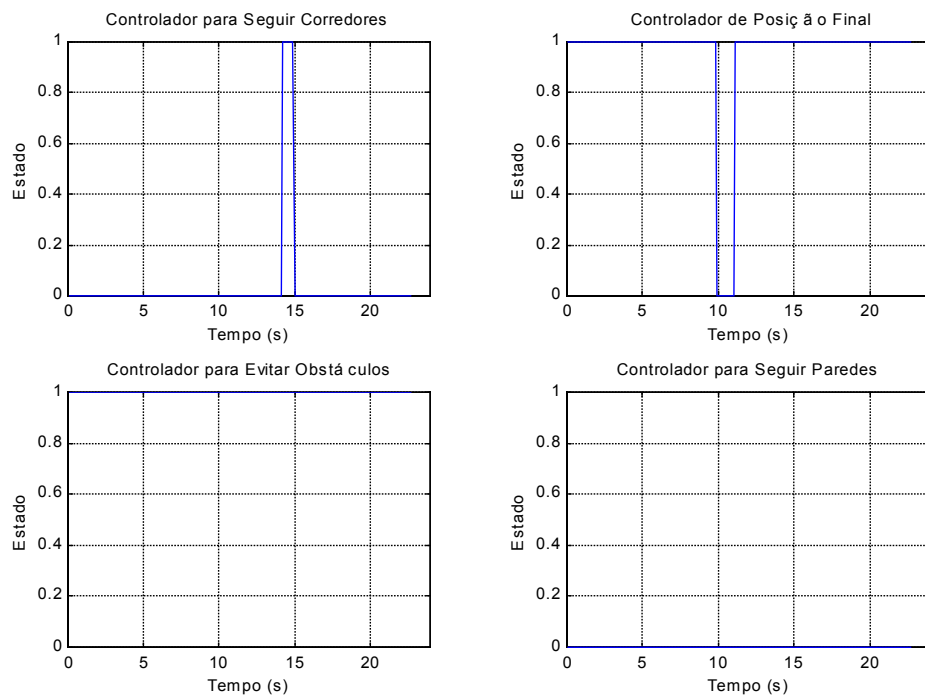


Figura 46 Estados dos controladores (Simulação #1).



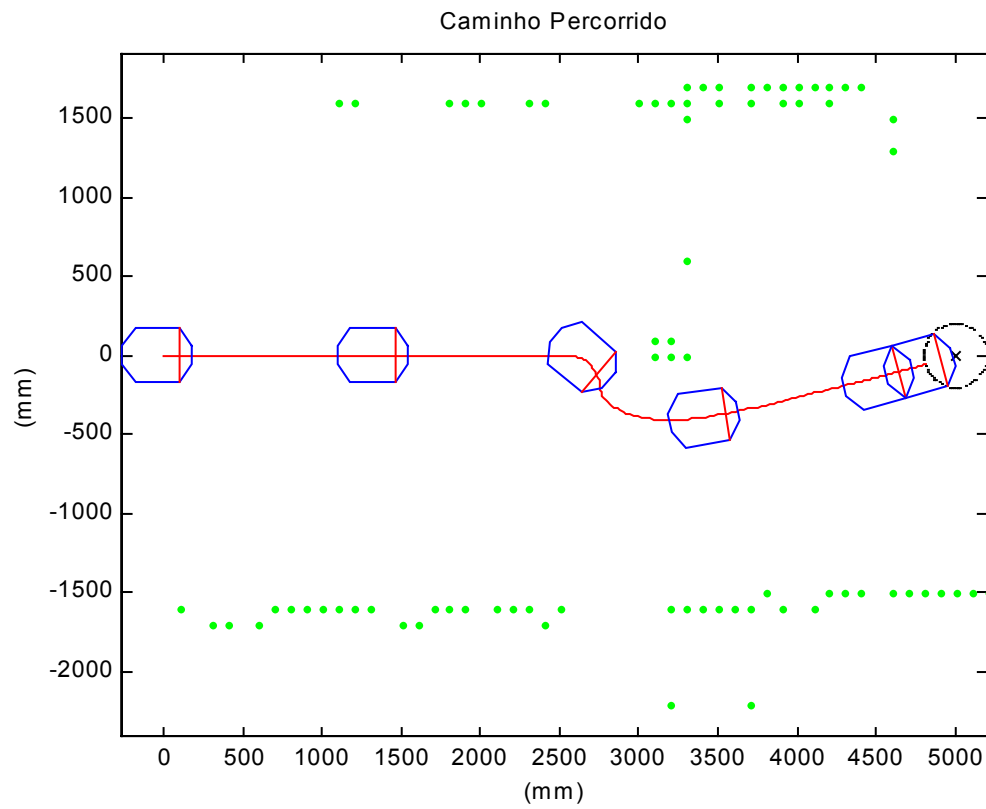


Figura 47 Caminho percorrido pelo robô (Simulação #1).

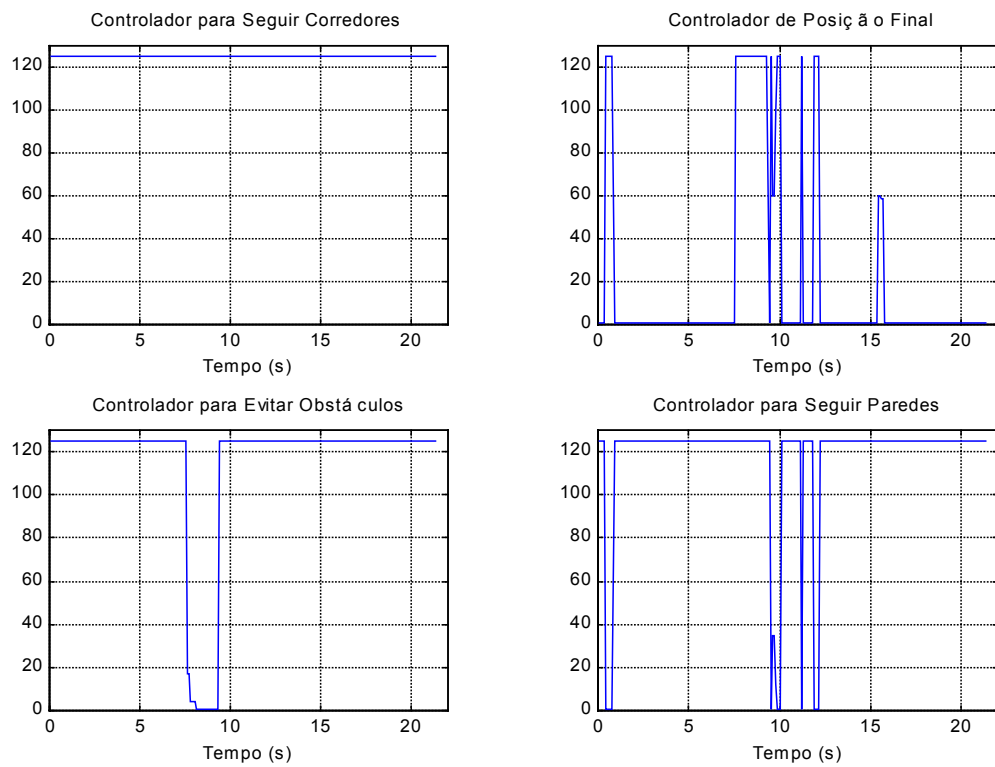


Figura 48 Covariâncias associadas aos controladores (Experimento #1).

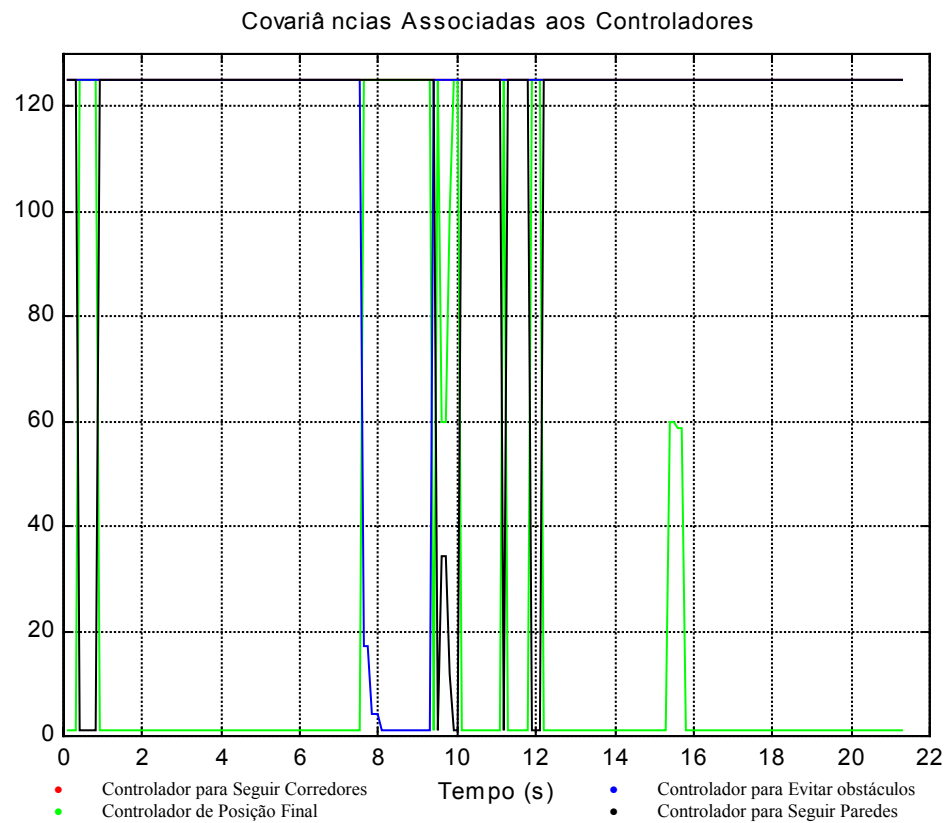


Figura 49 Covariâncias associadas aos controladores sobrepostas (Experimento #1).

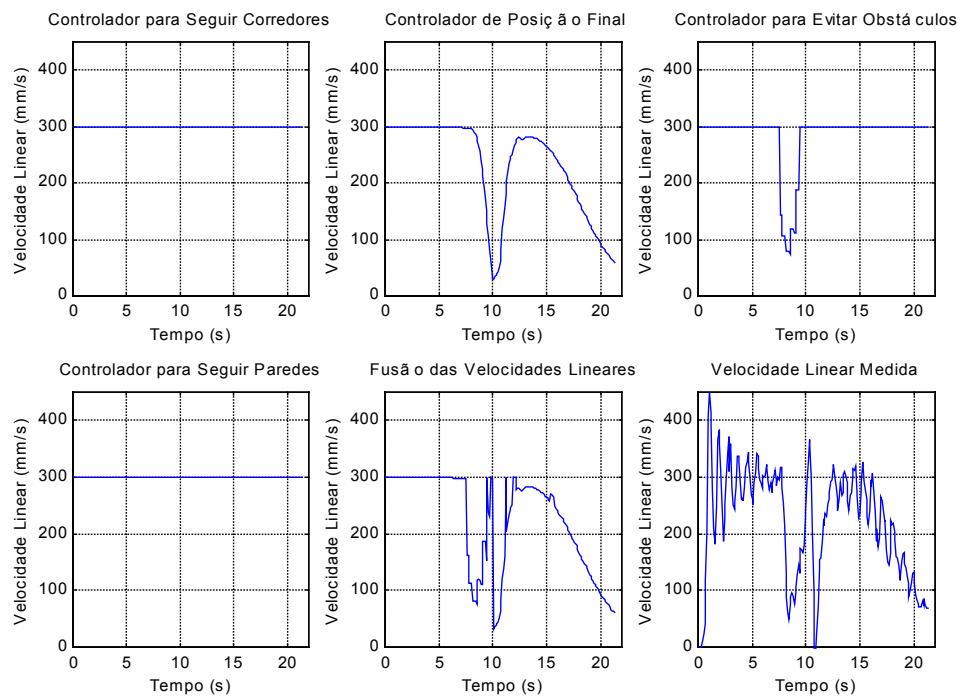


Figura 50 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento #1).

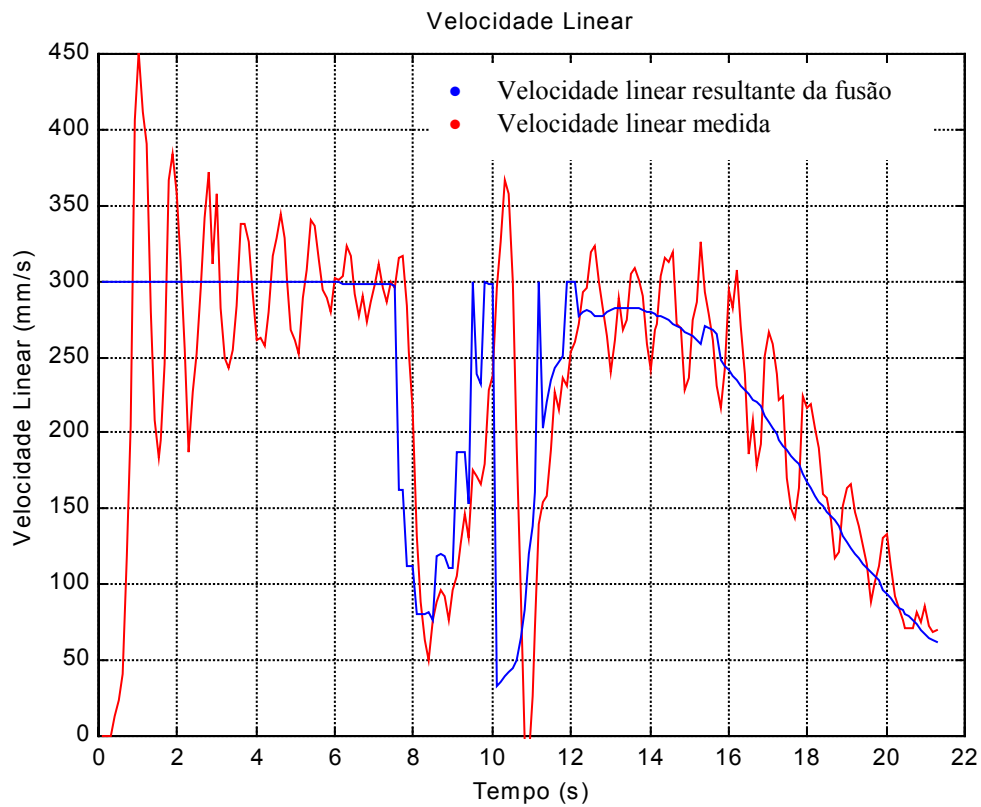


Figura 51 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento #1).

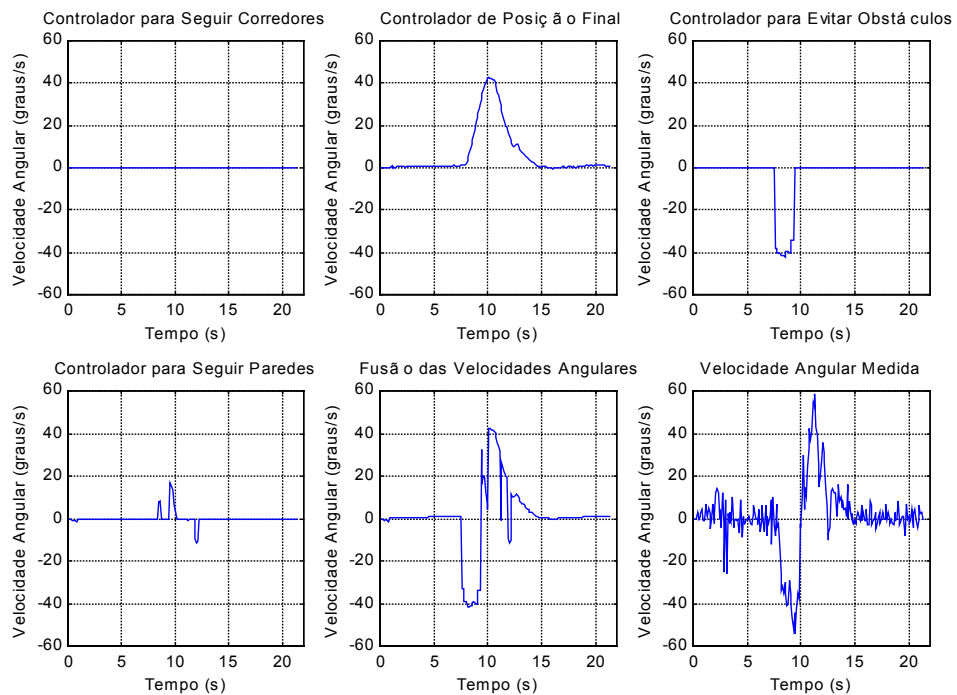


Figura 52 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #1).

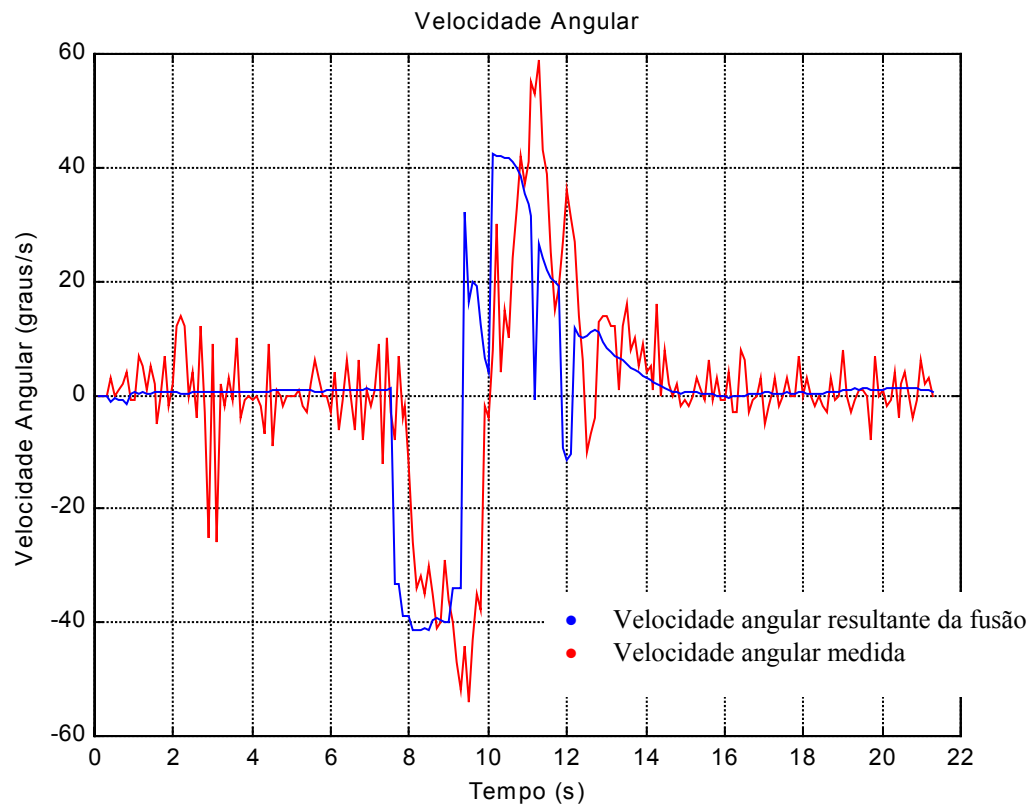


Figura 53 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #1).

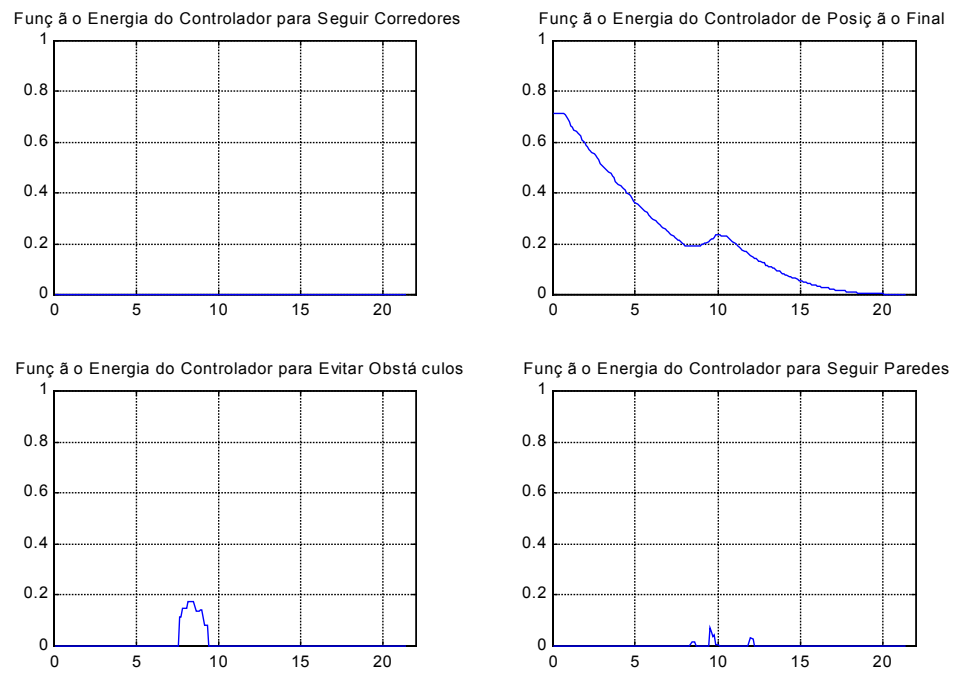


Figura 54 Funções energia associadas aos quatro controladores envolvidos no processo (Experimento #1).

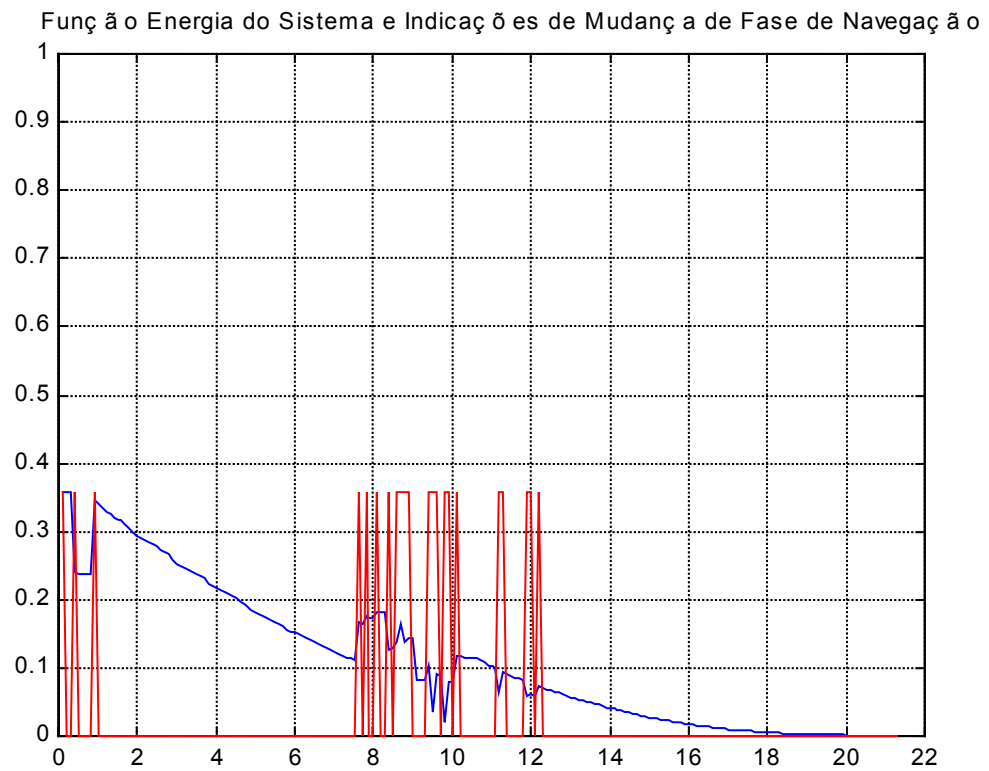


Figura 55 Função energia do sistema e indicações de mudanças de fase de navegação (Experimento #1).

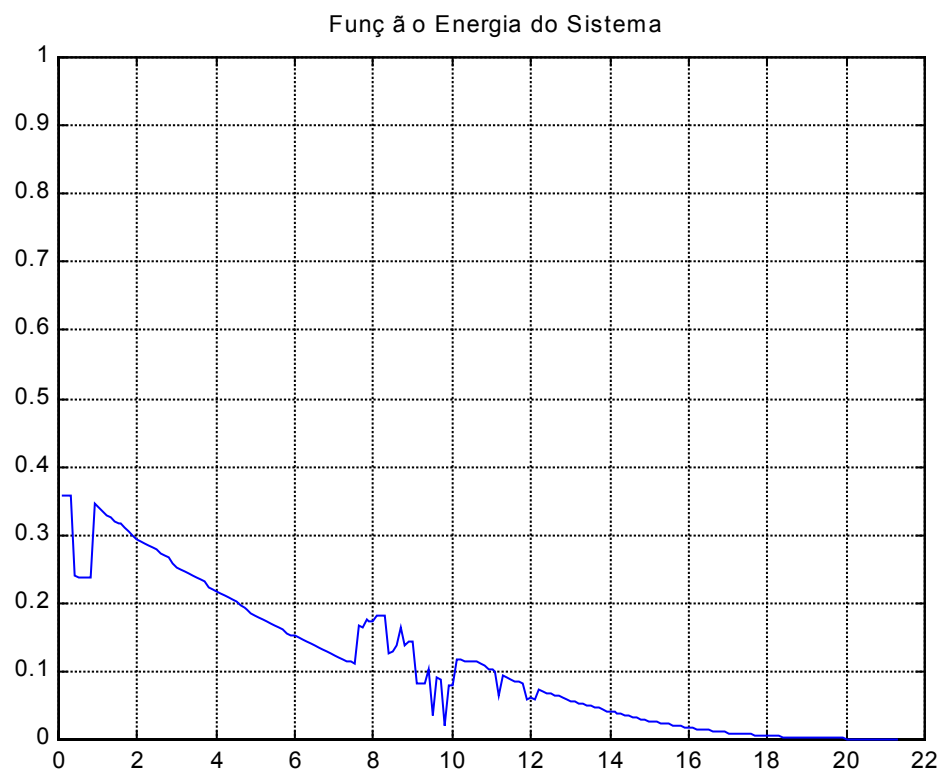


Figura 56 Função energia do sistema (Experimento #1).

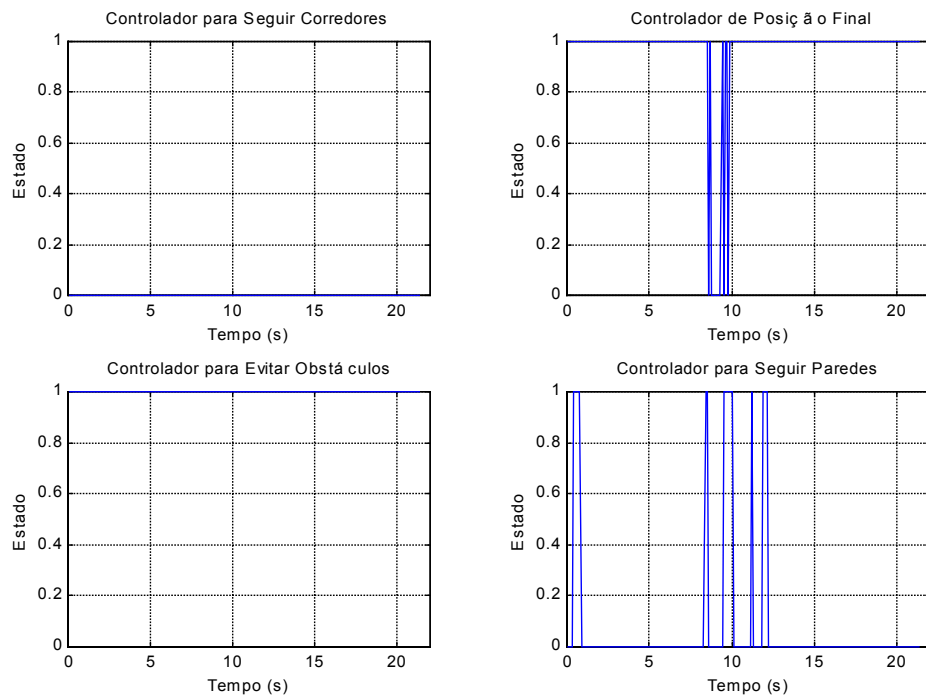


Figura 57 Estados dos controladores (Experimento #1).

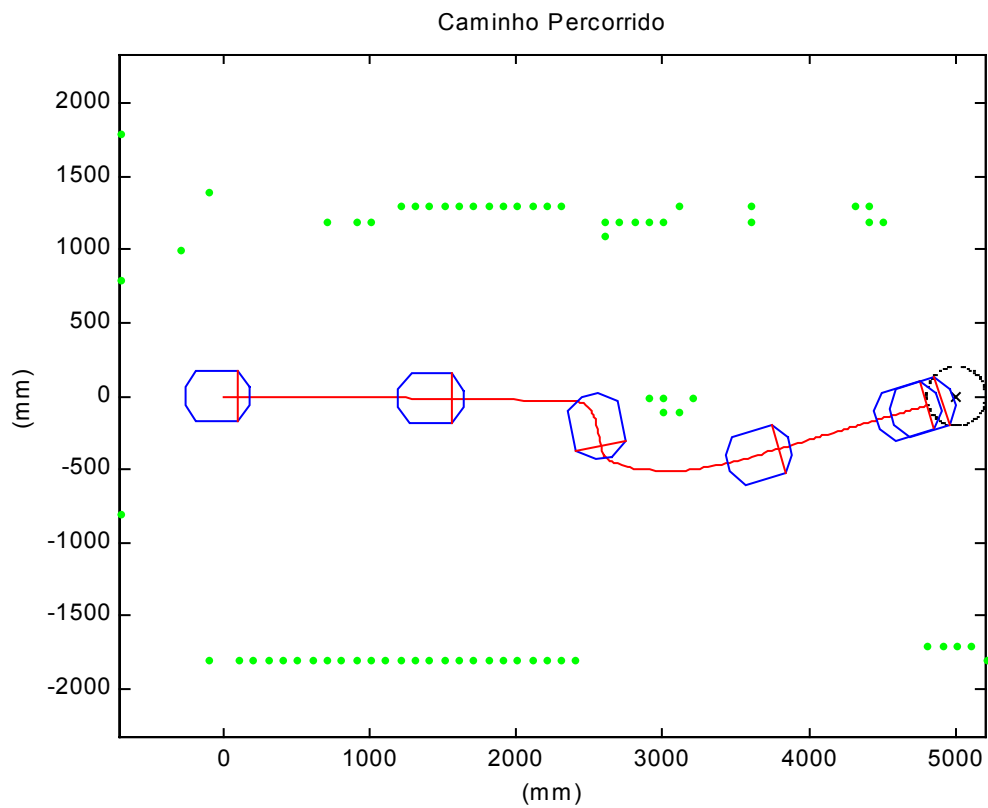


Figura 58 Caminho percorrido pelo robô (Experimento #1).

### 6.2.2 Teste #2

Neste teste, o robô é posicionado em um corredor com três metros de largura e deve chegar a um ponto localizado cinco metros diretamente à sua frente. No entanto, no meio do trajeto entre o ponto de partida e o ponto de destino é posicionado um obstáculo em forma de V. O objetivo deste teste é verificar como a arquitetura de controle proposta se porta ao enfrentar problemas de mínimos locais, como é o caso para o obstáculo em forma de V em questão.

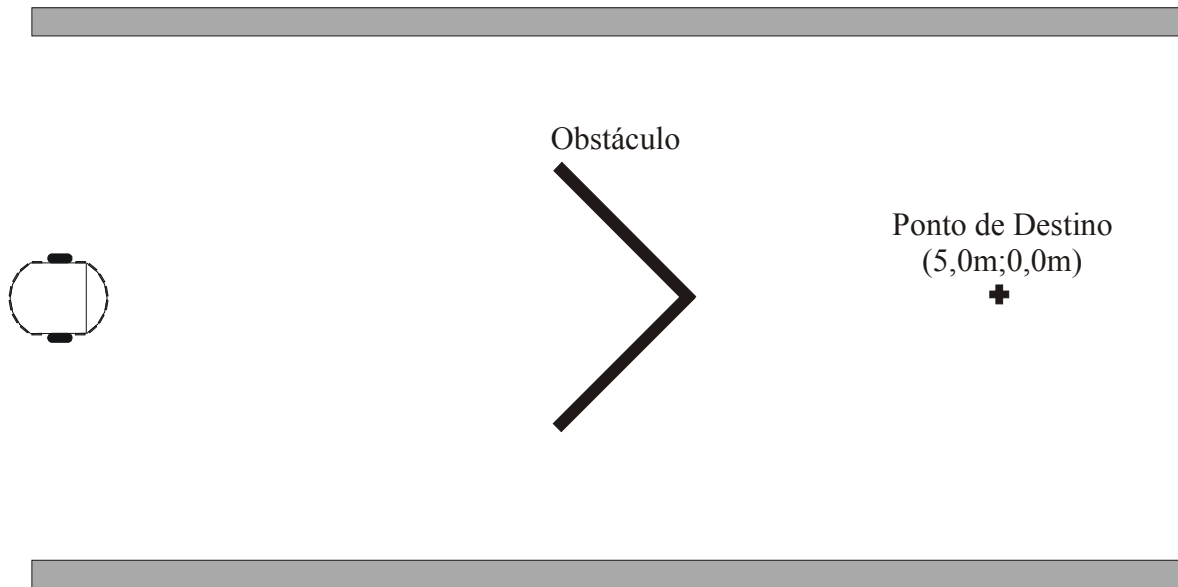


Figura 59 Teste #2.

Tabela 8 Índices de Avaliação de Desempenho do Teste #2.

Índice	Simulação	Experimento	Valores Ideais
Segurança	209 mm	170 mm	1250 mm
Velocidade Média	204 mm/s	208 mm/s	300 mm/s
Suavidade	1,21°	1,27°	0°
Distância Percorrida	5,46 m	5,36 m	5,00 m
Tempo	26,70 s	25,80 s	16,67 s

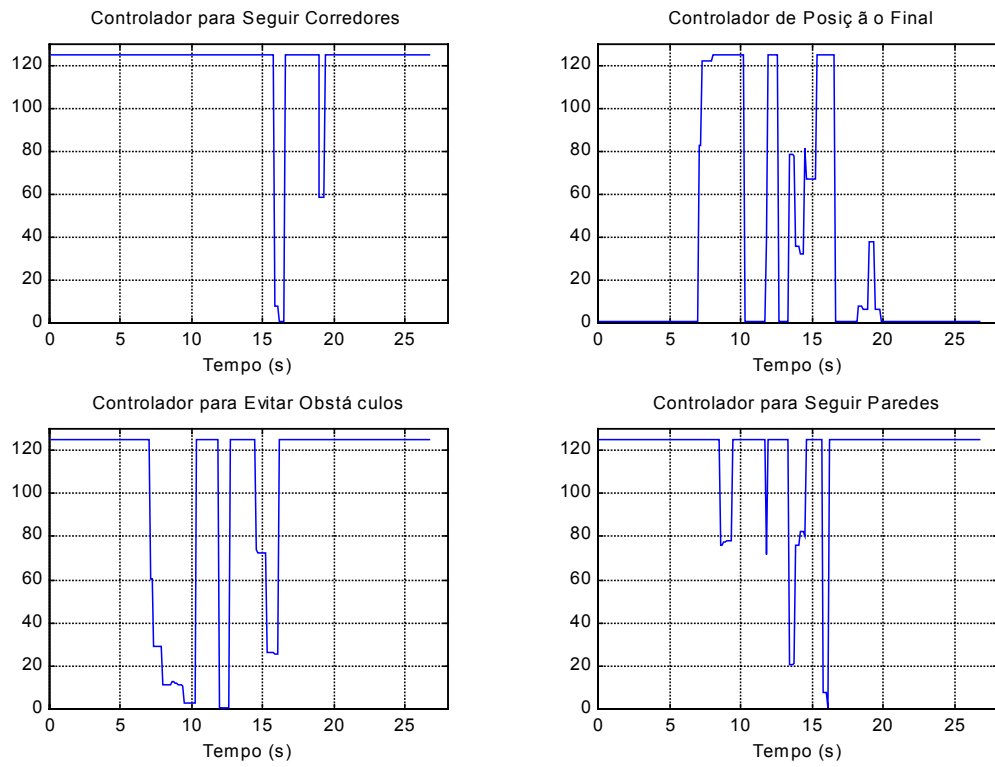


Figura 60 Covariâncias associadas aos controladores (Simulação #2).

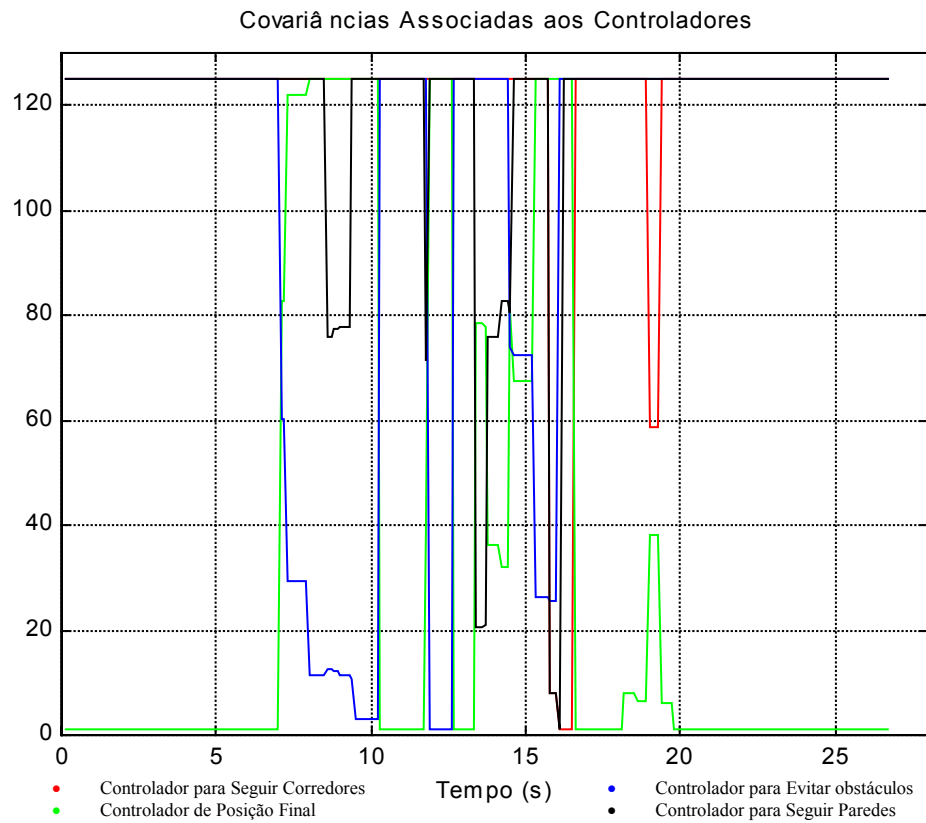


Figura 61 Covariâncias associadas aos controladores sobrepostas (Simulação #2).



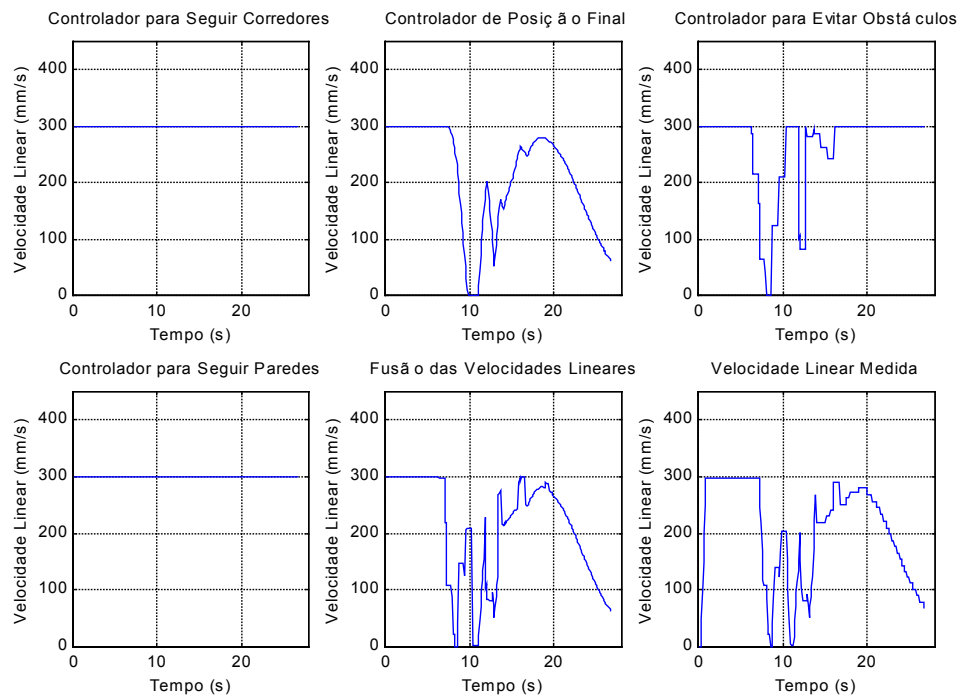


Figura 62 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #2).

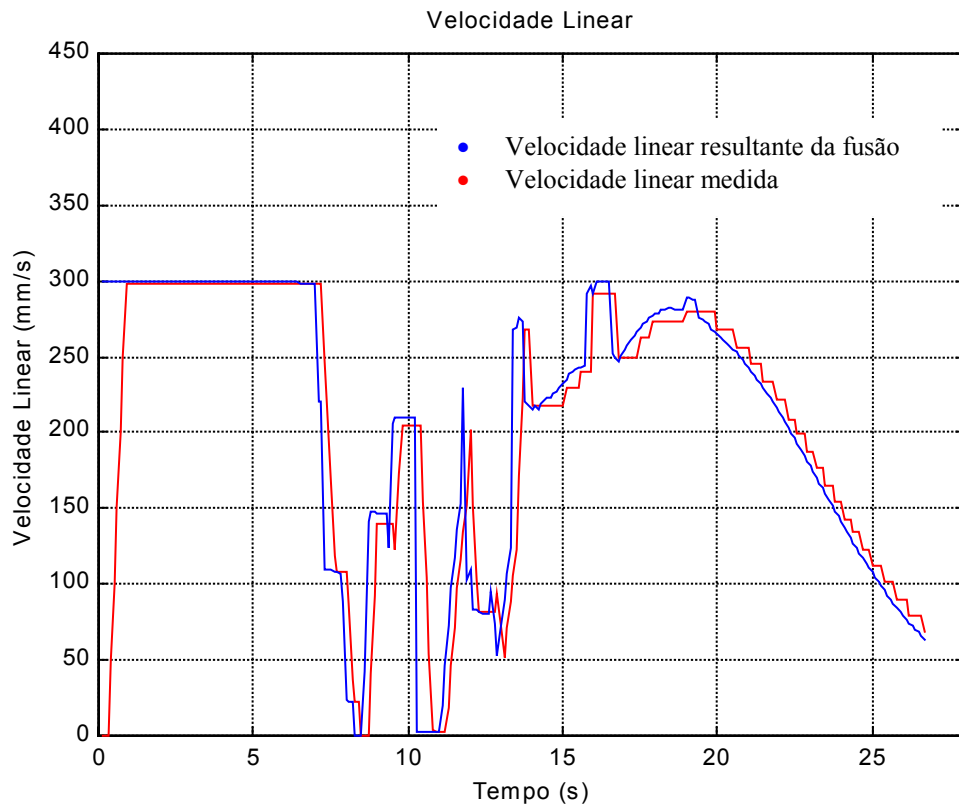


Figura 63 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #2).

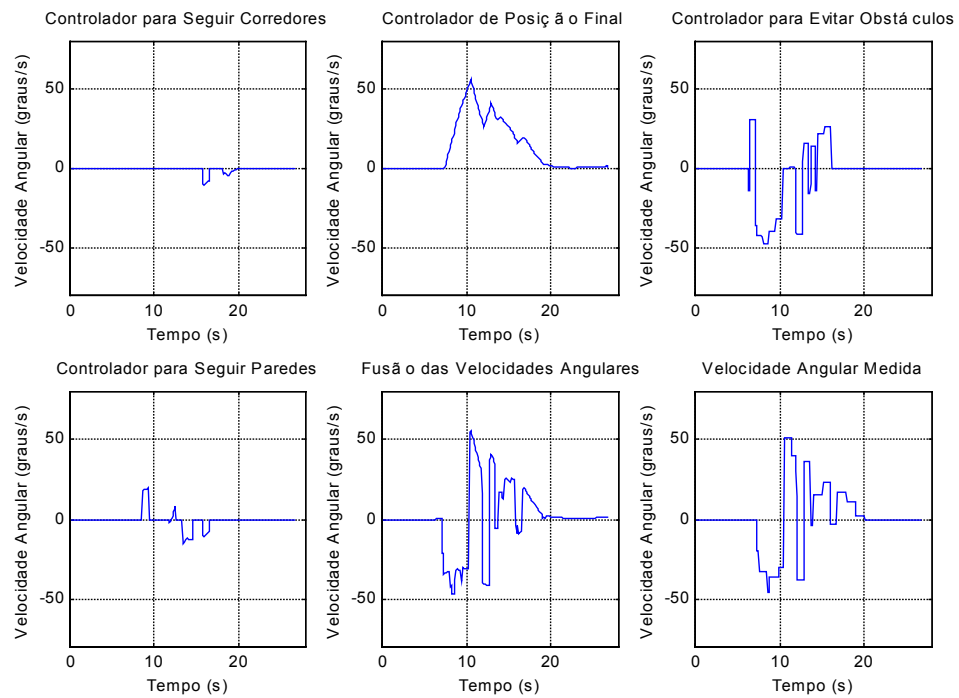


Figura 64 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #2).

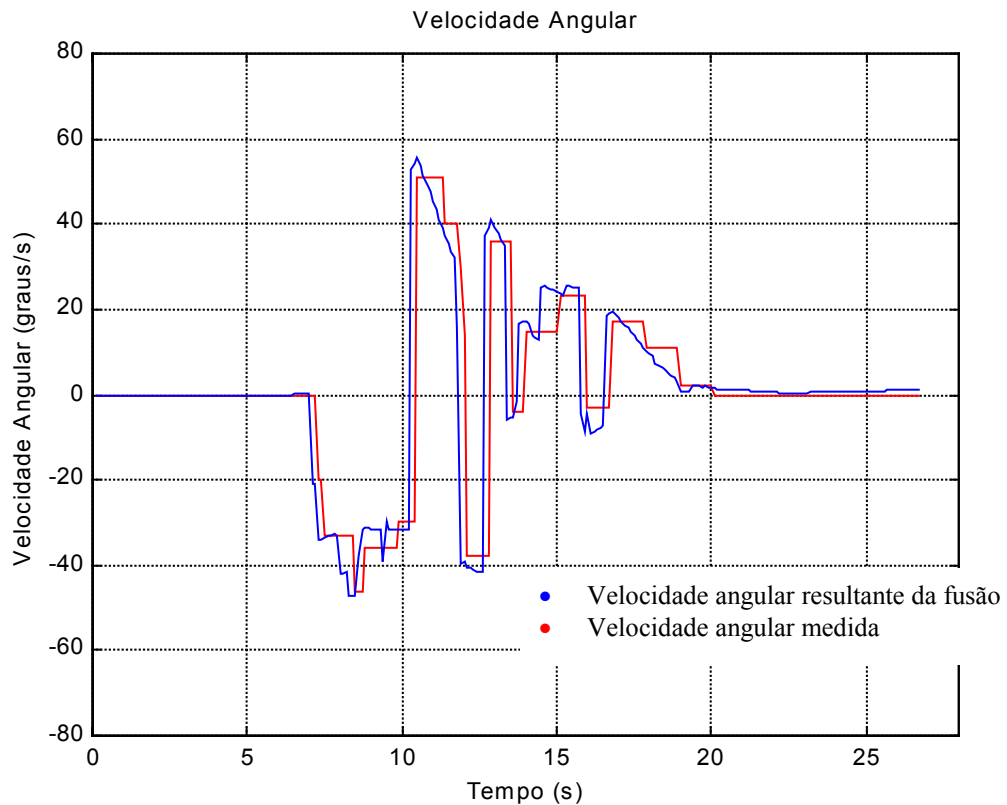


Figura 65 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Simulação #2).

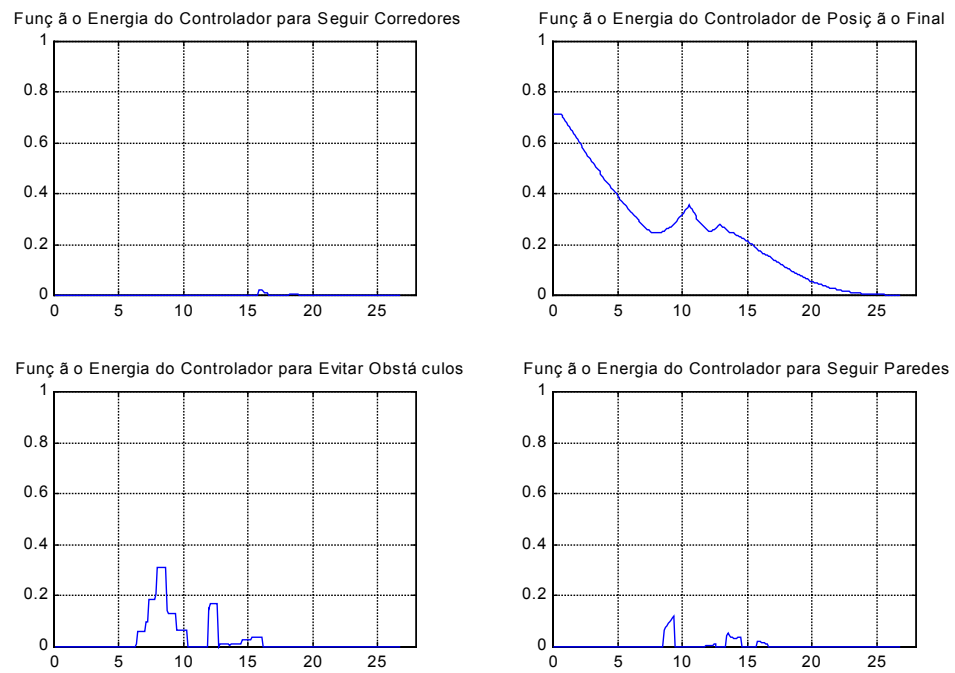


Figura 66 Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #2).

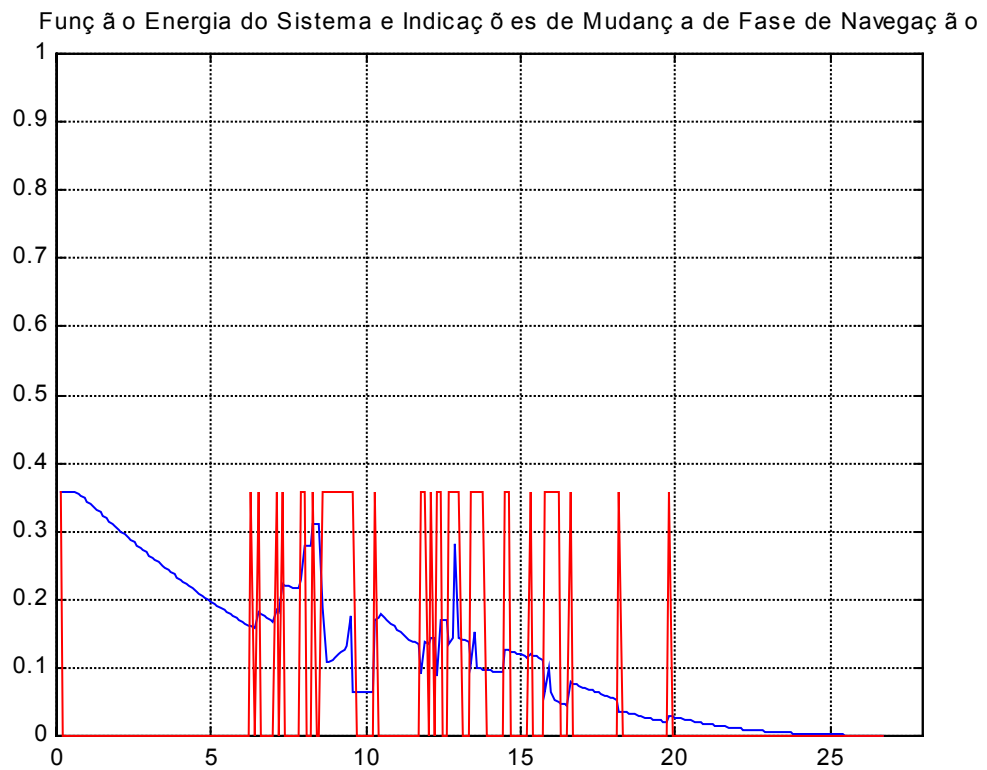


Figura 67 Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #2).

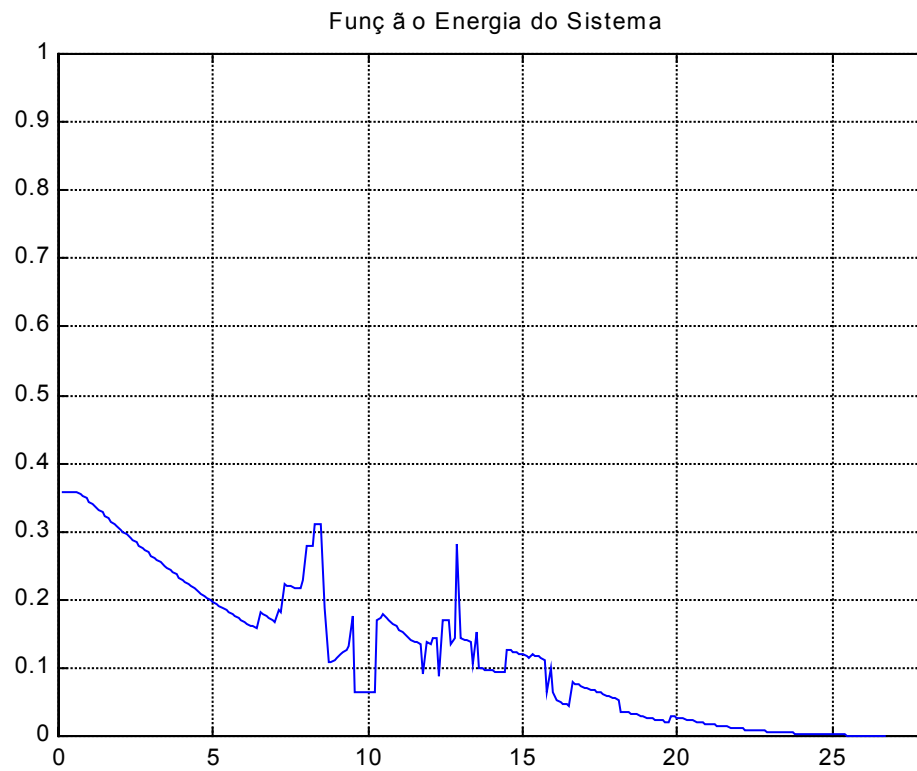


Figura 68 Função energia do sistema (Simulação #2).

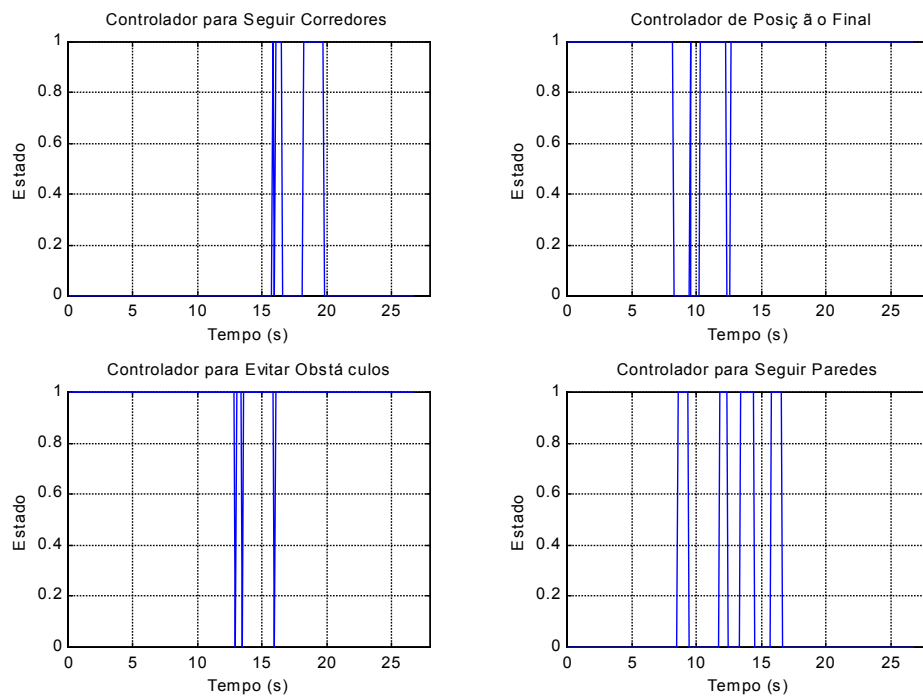


Figura 69 Estados dos controladores (Simulação #2).

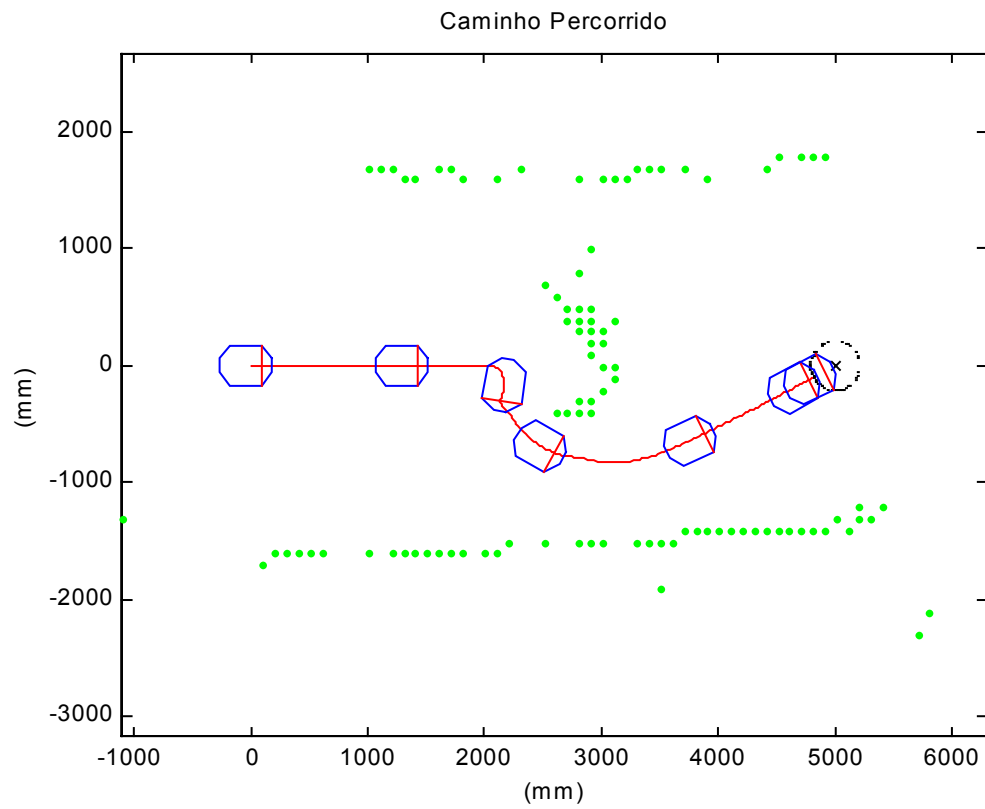


Figura 70 Caminho percorrido pelo robô (Simulação #2).

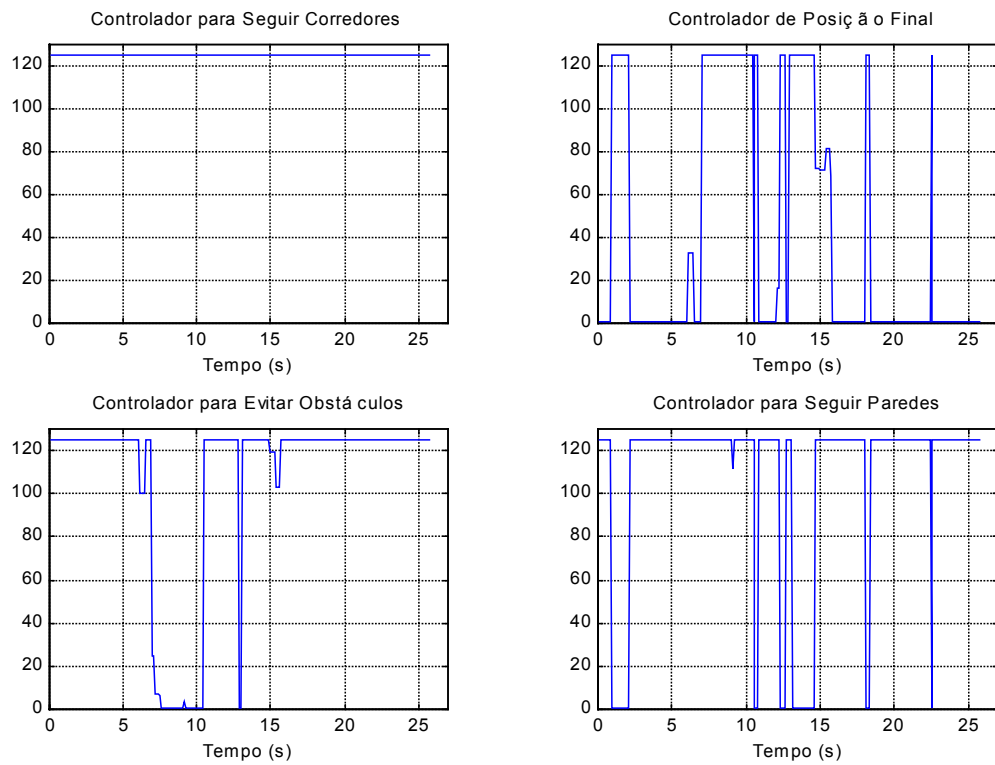


Figura 71 Covariâncias associadas aos controladores (Experimento 2).

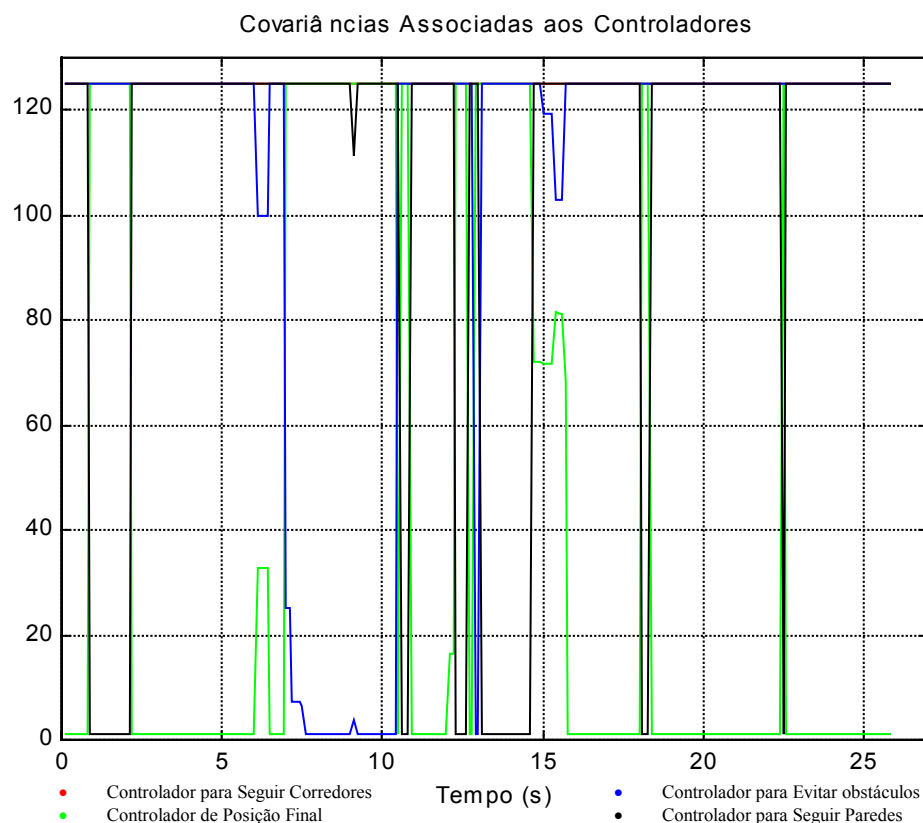


Figura 72 Covariâncias associadas aos controladores sobrepostas (Experimento 2).

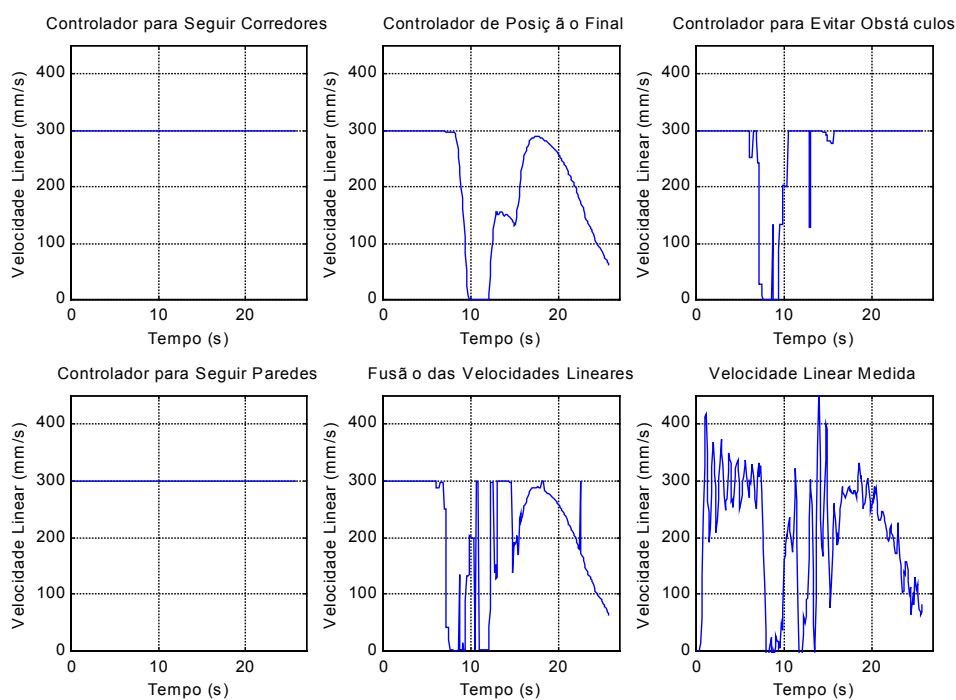


Figura 73 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento 2).

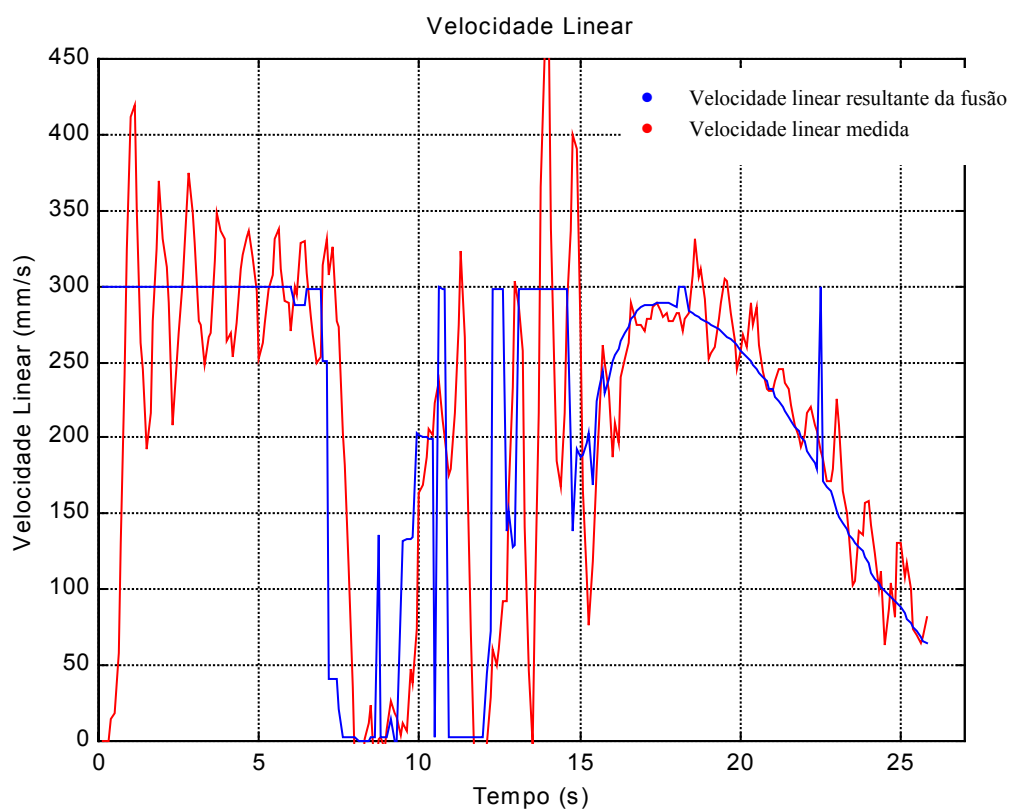


Figura 74 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento 2).

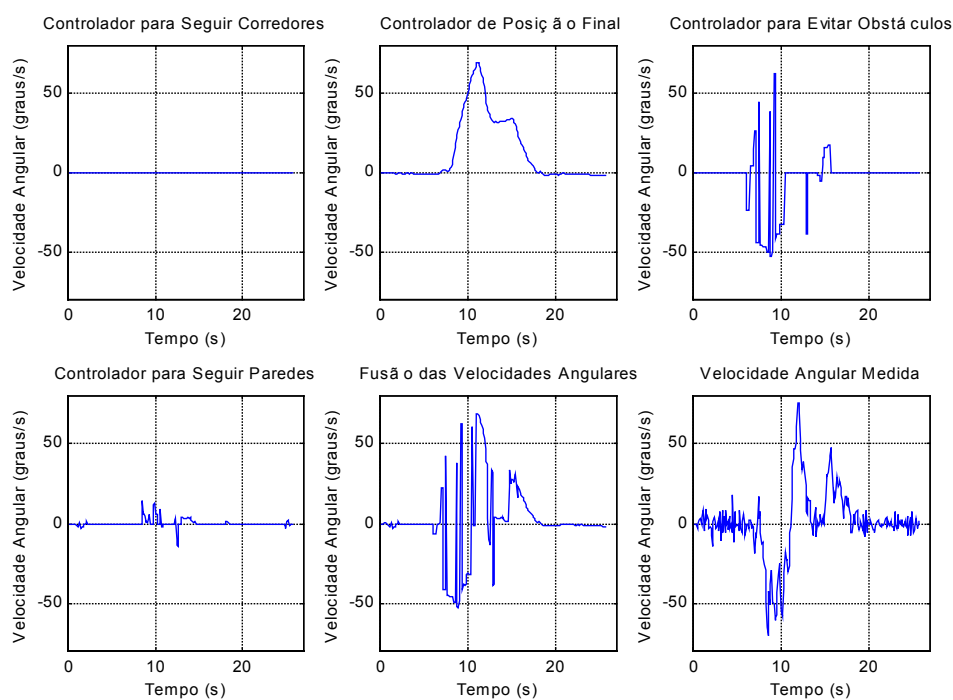


Figura 75 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #2).

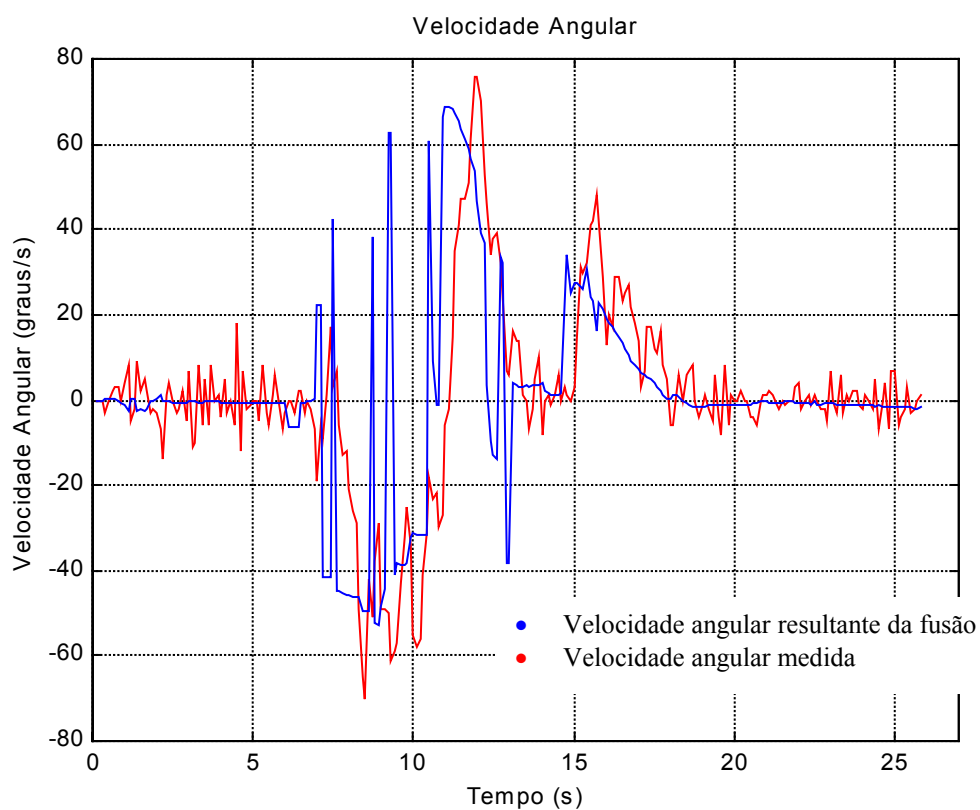


Figura 76 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #2).

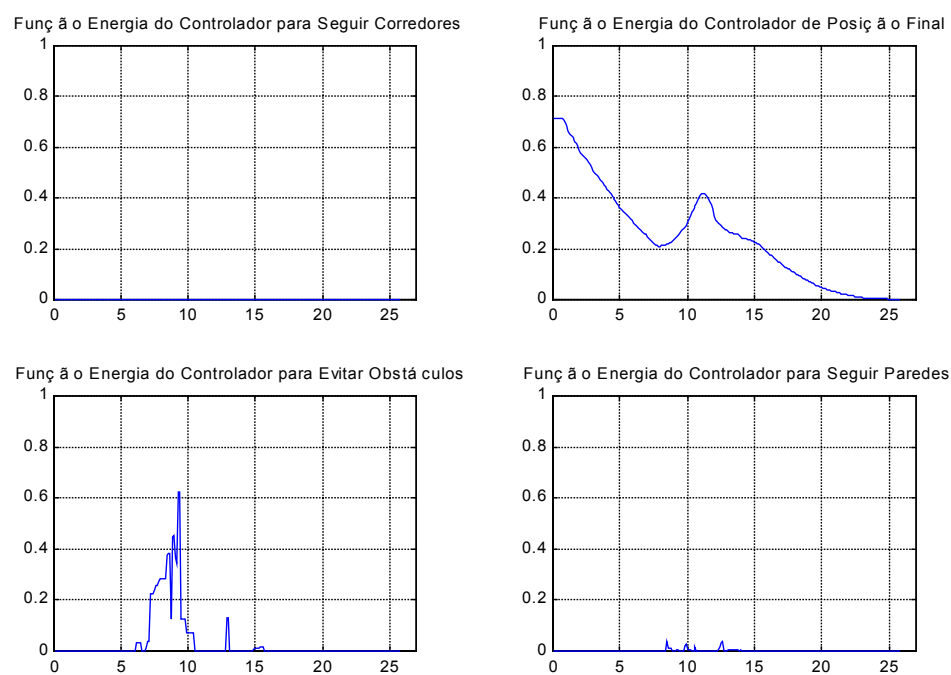


Figura 77 Funções energia associadas aos quatro controladores envolvidos no processo (Experimento 2).



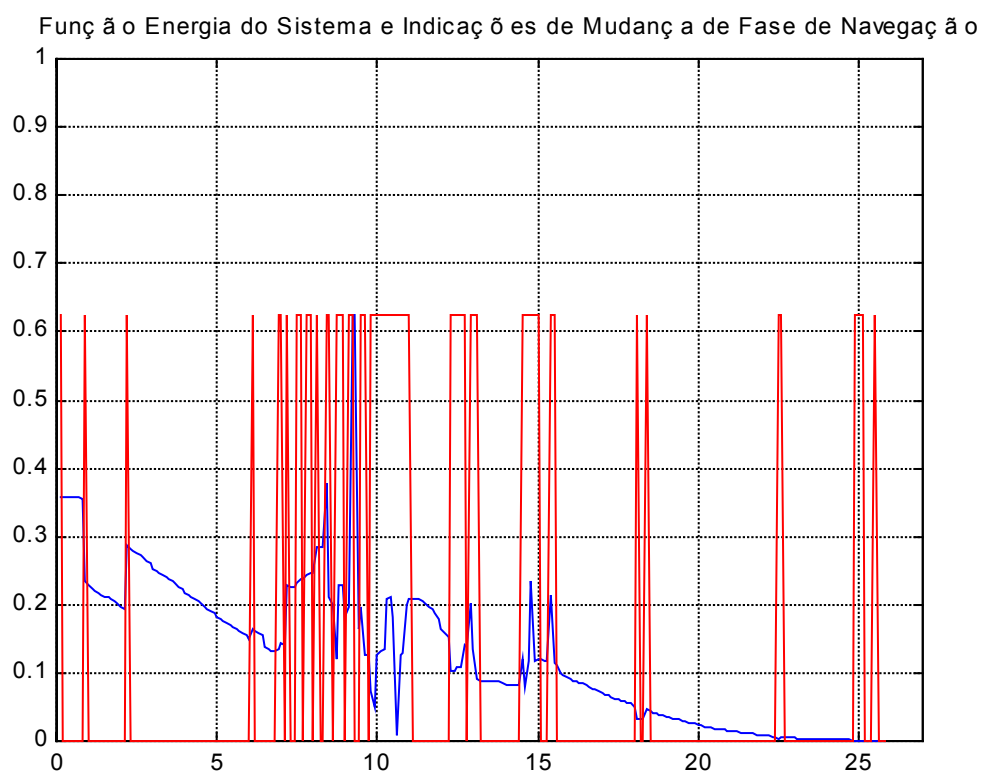


Figura 78 Função energia do sistema e indicações de mudanças de fase de navegação (Experimento 2).

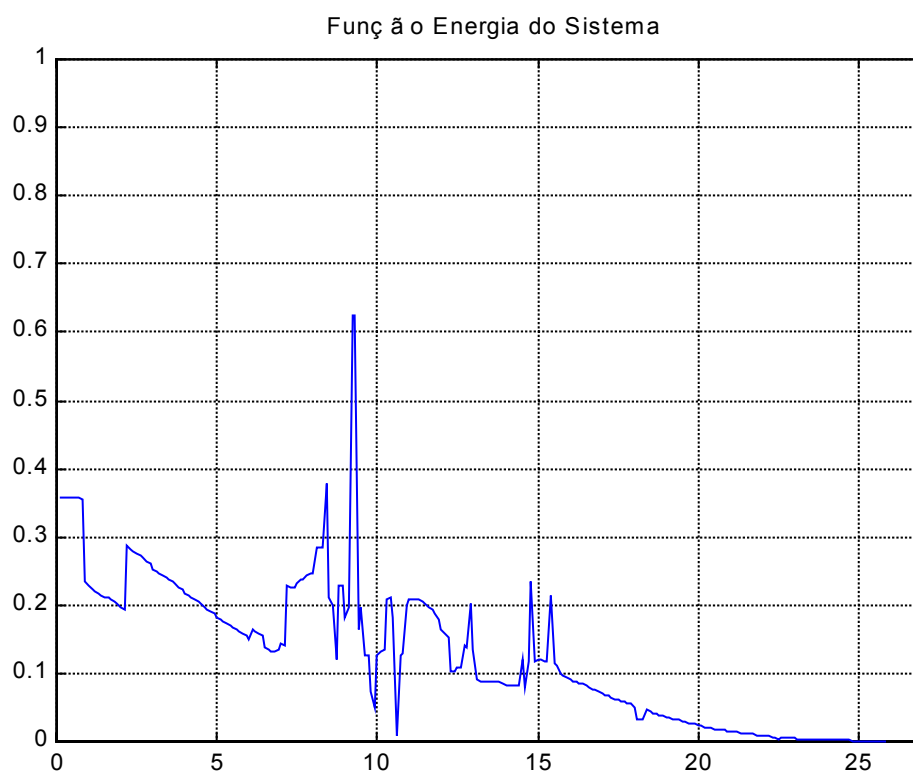


Figura 79 Função energia do sistema (Experimento 2).

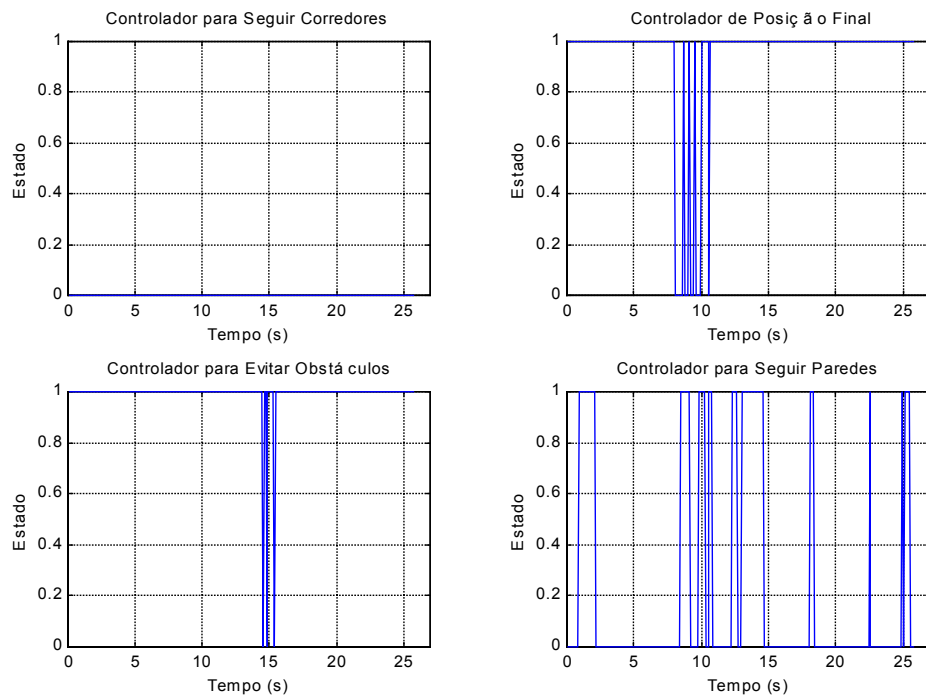


Figura 80 Estados dos controladores (Experimento 2).

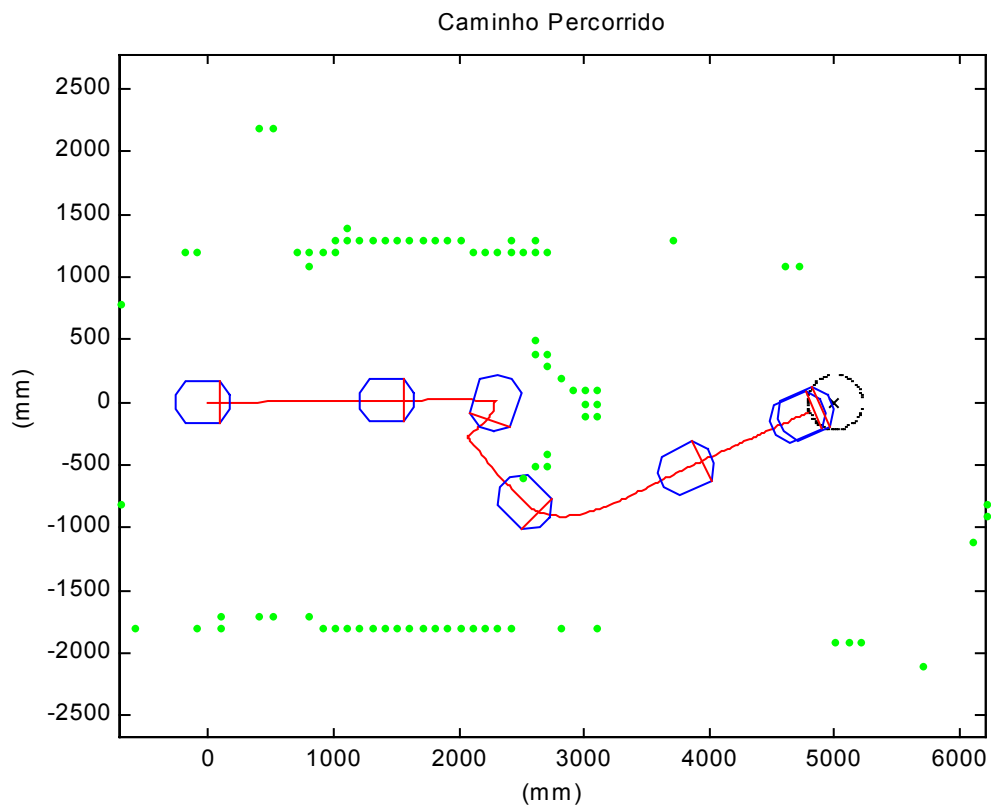


Figura 81 Caminho percorrido pelo robô (Experimento 2).

### 6.2.3 Teste #3

O objetivo deste teste consiste em guiar o robô de um ponto inicial (coordenadas [0m, 0m]) a um ponto de destino (coordenadas [12m, 5m]) em um ambiente composto por corredores, portas, paredes e obstáculos. Enquanto busca o ponto de destino, o robô deve navegar ao longo de corredores e evitar os obstáculos que aparecem em seu caminho. O ponto de partida e o ponto de destino não estão no mesmo corredor, e para o robô chegar ao seu objetivo, o caminho mais curto assemelha-se à forma de um “Z”.

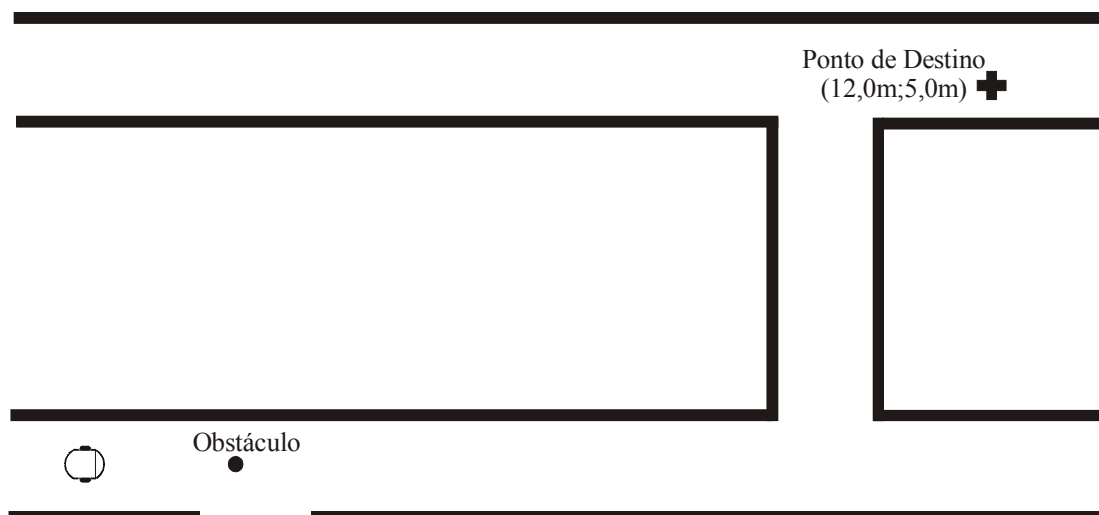


Figura 82 Teste #3.

Tabela 9 Índices de Avaliação de Desempenho do Teste #3.

Índice	Simulação	Experimento	Valores Ideais
Segurança	222 mm	172 mm	500 mm
Velocidade Média	275 mm/s	293 mm/s	300 mm/s
Suavidade	0,60°	0,89°	0,32°
Distância Percorrida	16,57 m	16,14 m	17,00 m
Tempo	60,20 s	55,10 s	56,67 s

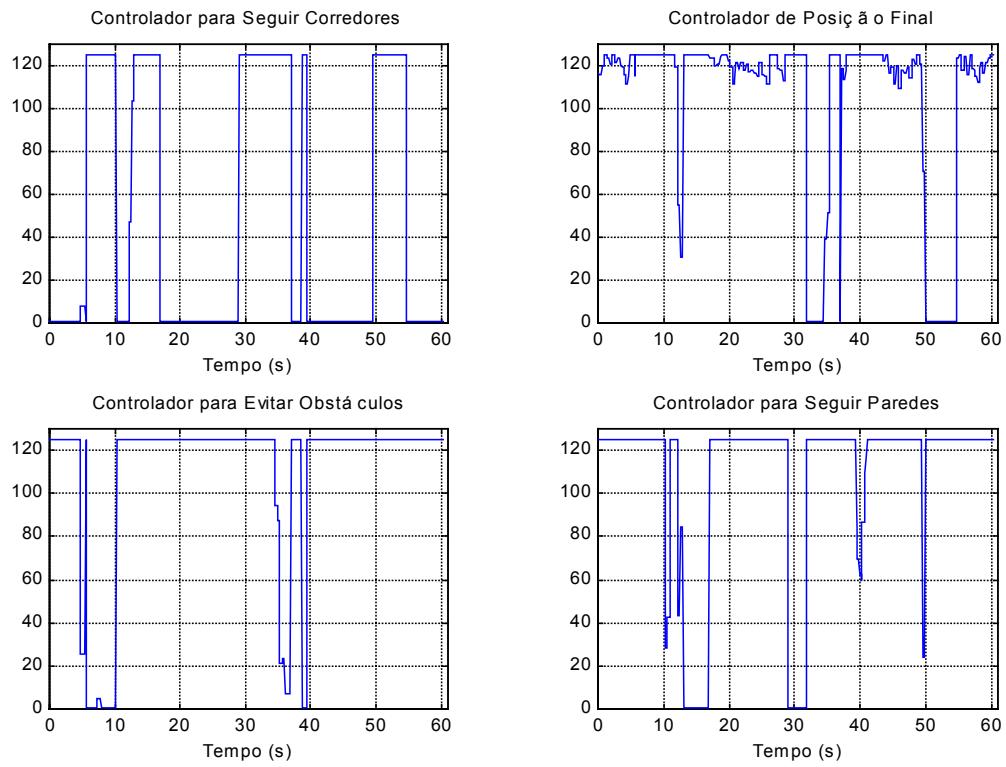


Figura 83 Covariâncias associadas aos controladores (Simulação #3).

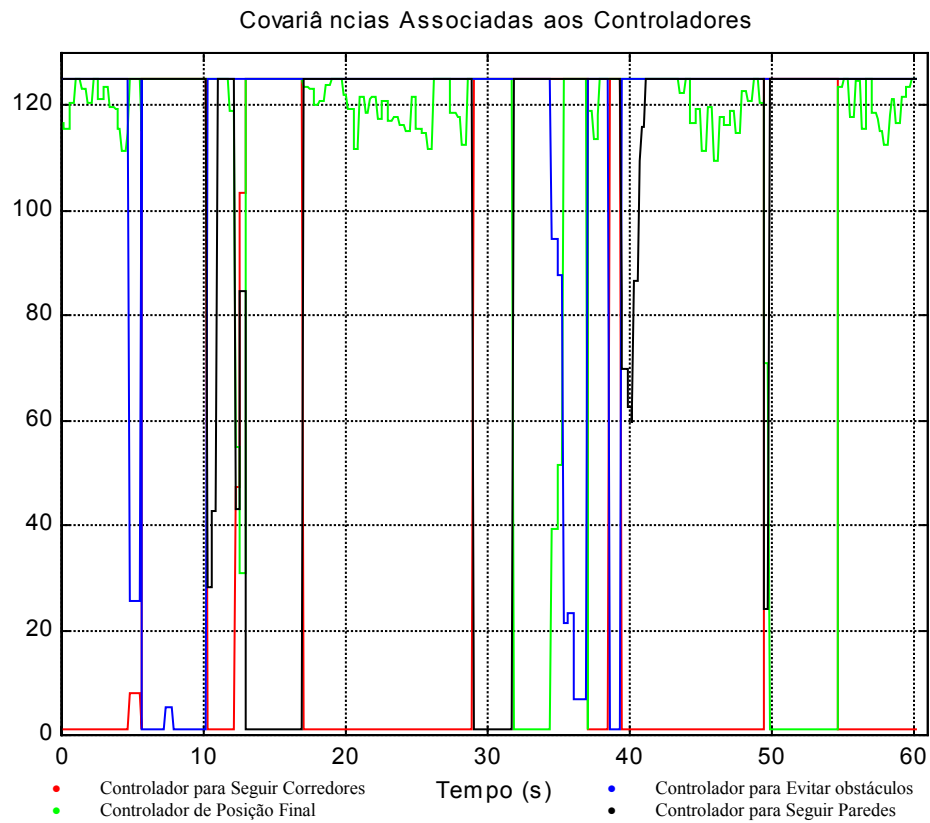


Figura 84 Covariâncias associadas aos controladores sobrepostas (Simulação #3).

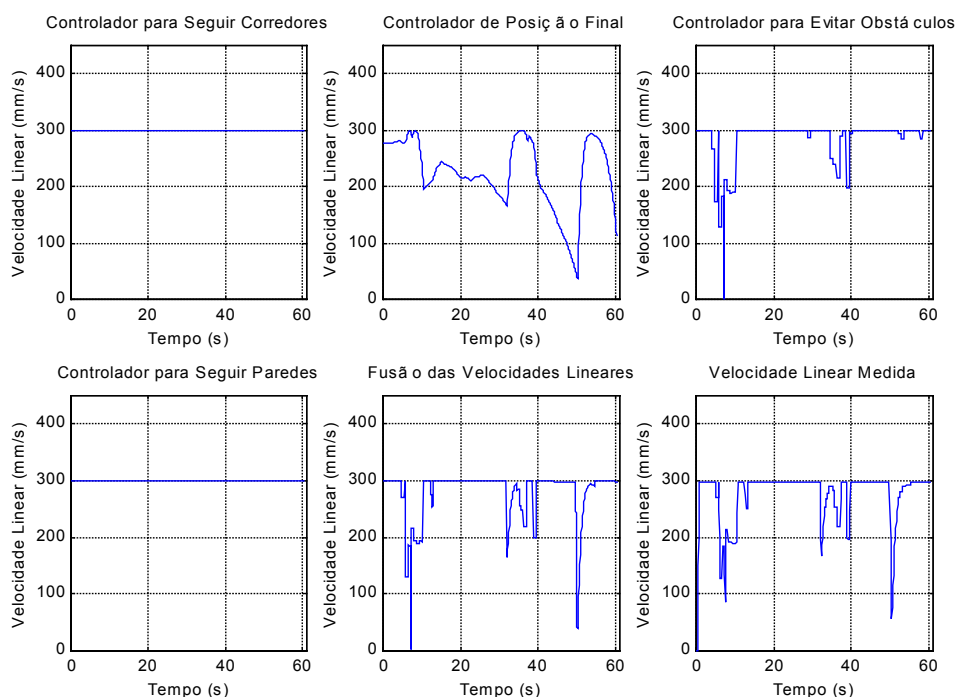


Figura 85 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #3).

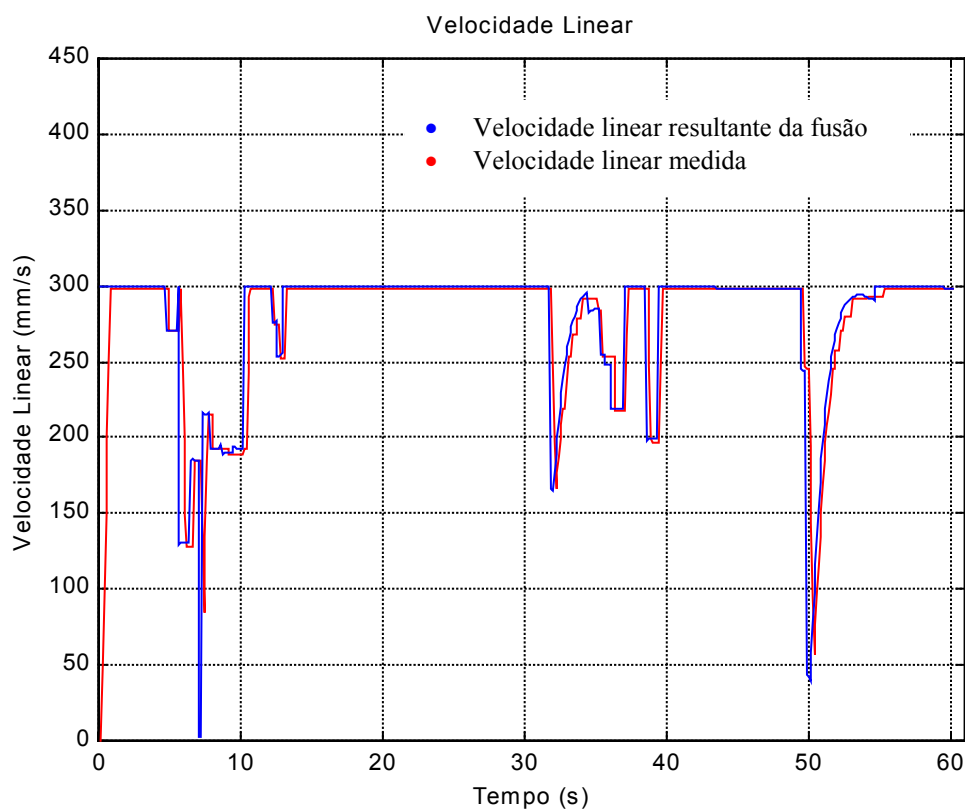


Figura 86 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #3).

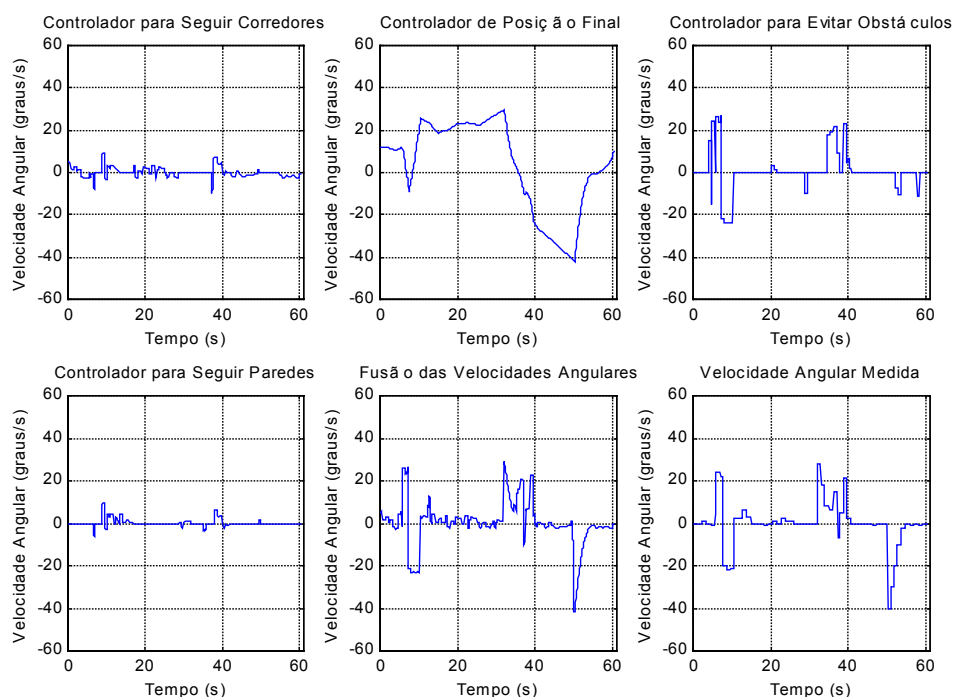


Figura 87 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #3).

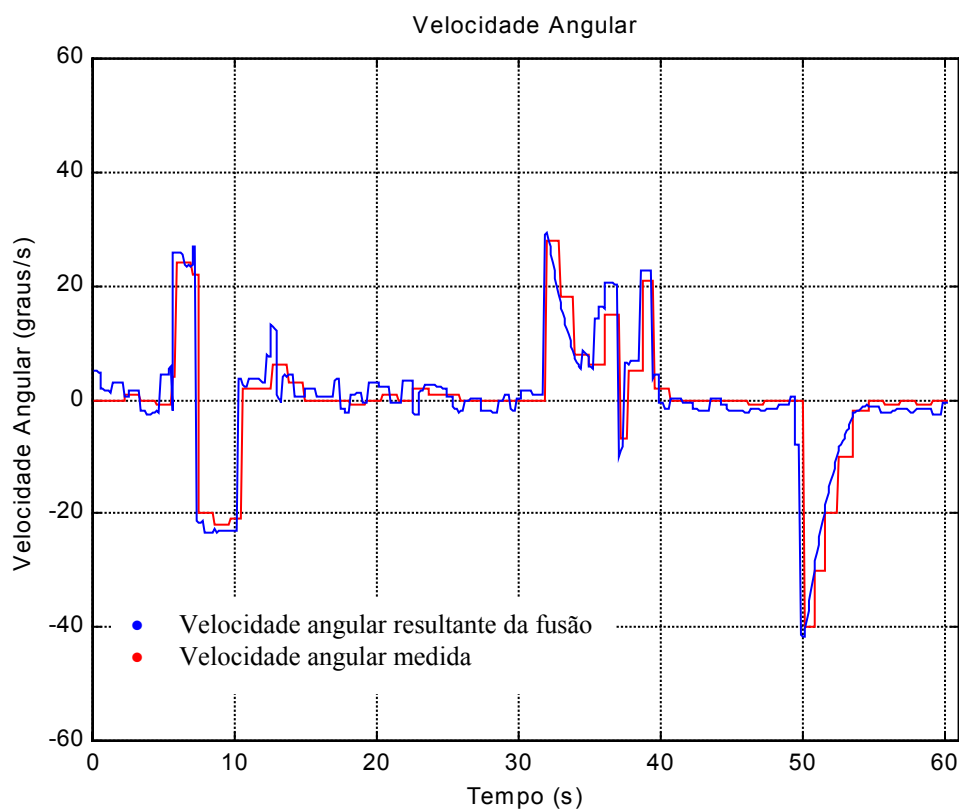


Figura 88 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Simulação #3).

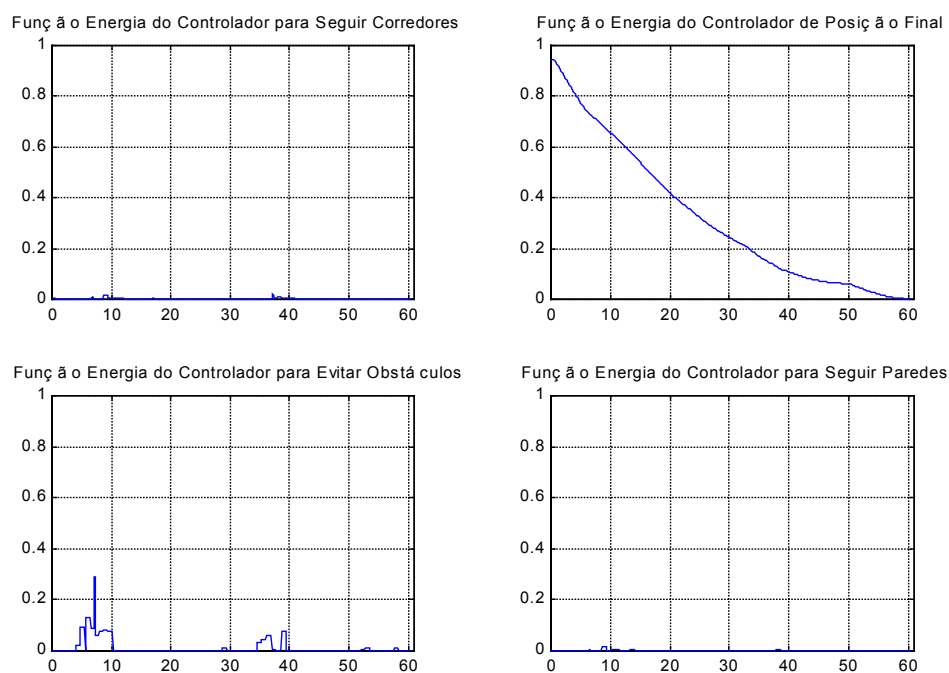


Figura 89 Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #3).

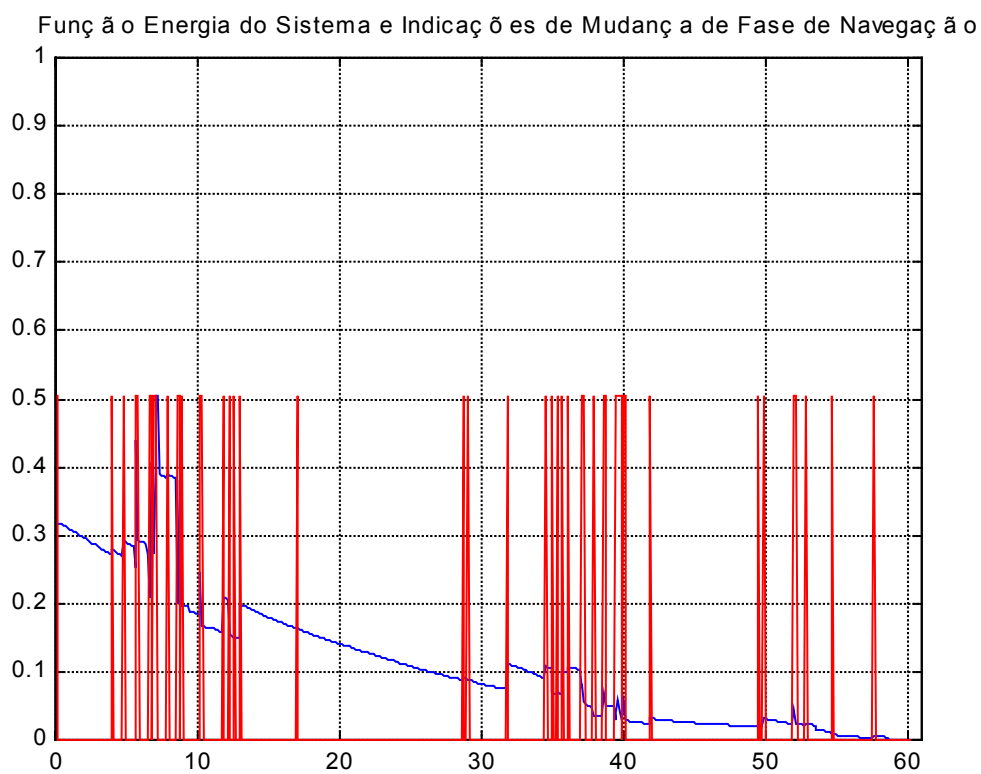


Figura 90 Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #3).

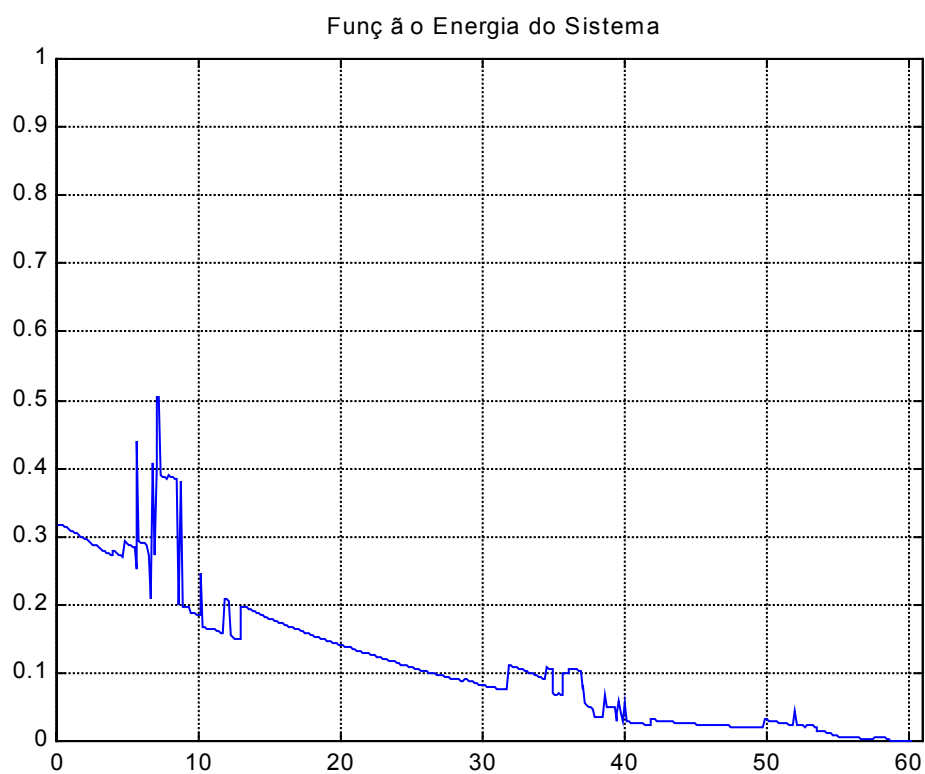


Figura 91 Função energia do sistema (Simulação #3).

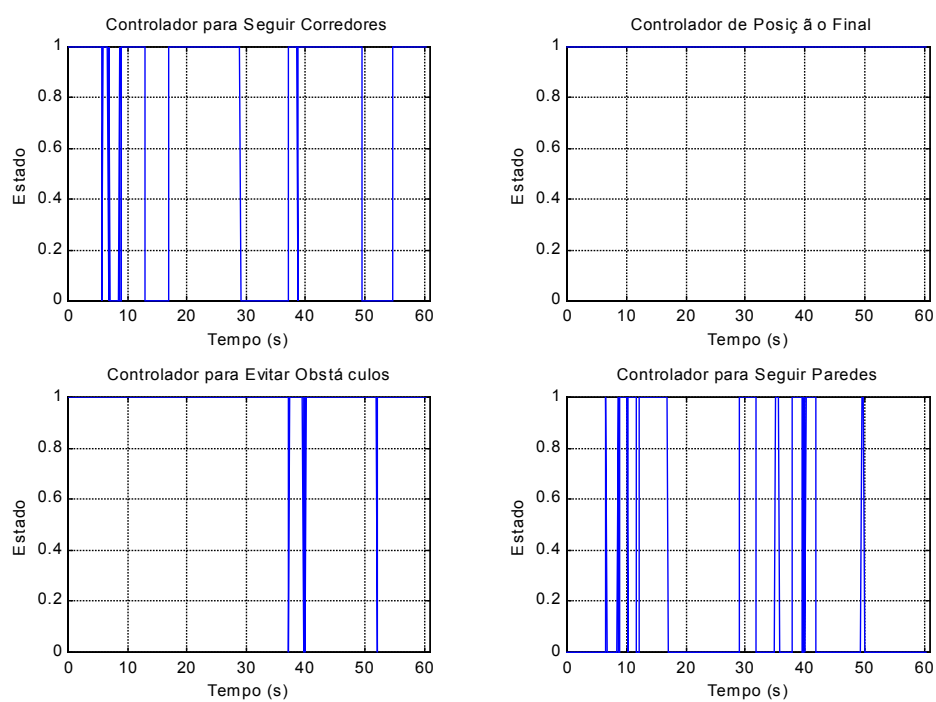


Figura 92 Estados dos controladores (Simulação #3).



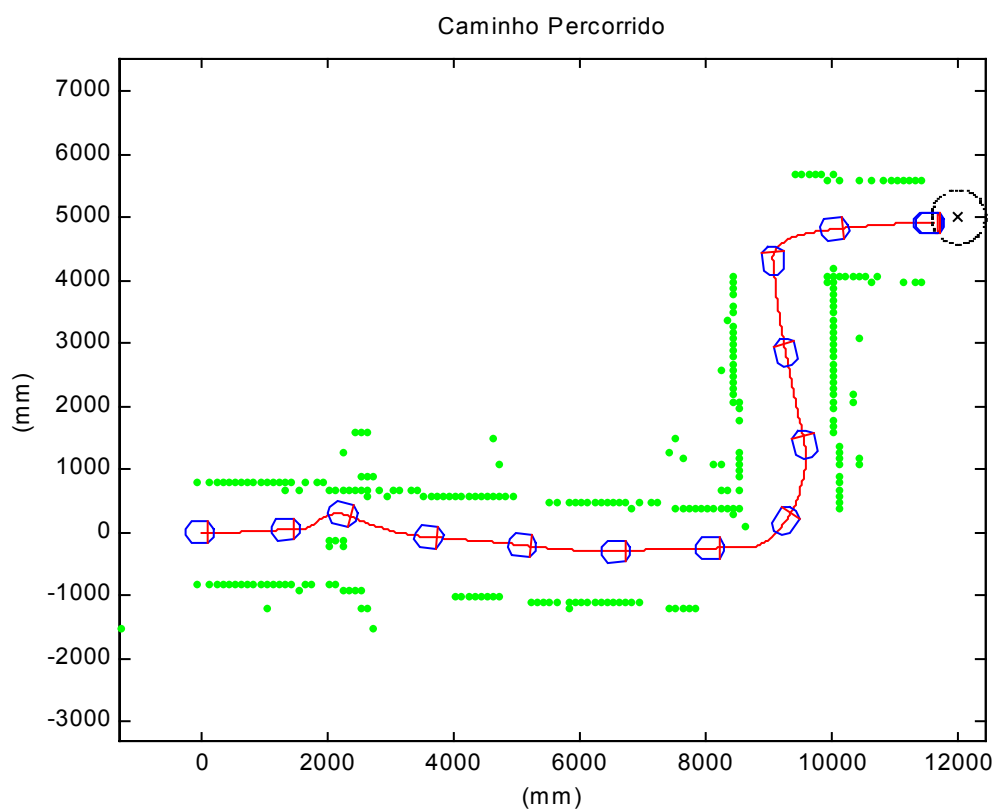


Figura 93 Caminho percorrido pelo robô (Simulação #3).

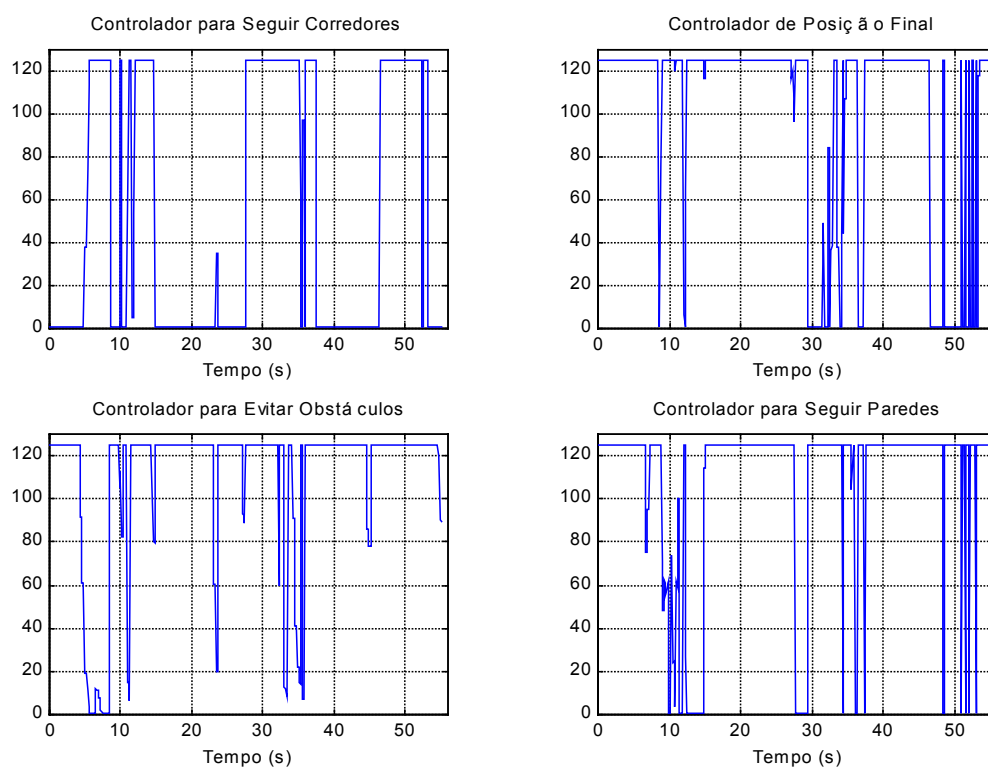


Figura 94 Covariâncias associadas aos controladores (Experimento #3).

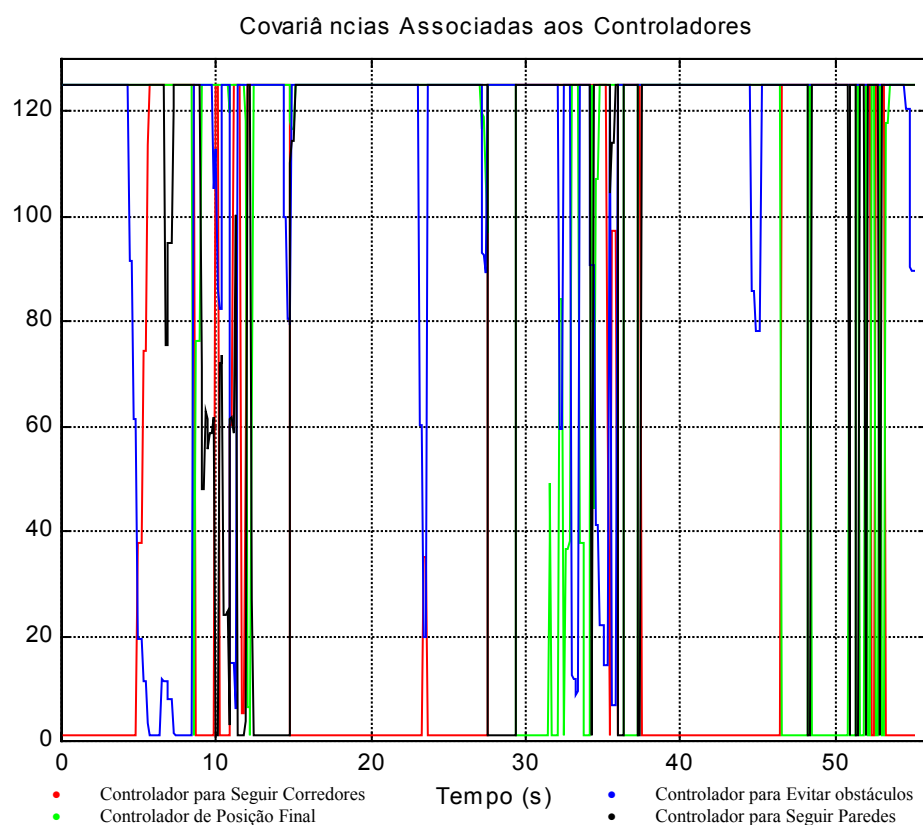


Figura 95 Covariâncias associadas aos controladores sobrepostas (Experimento #3).

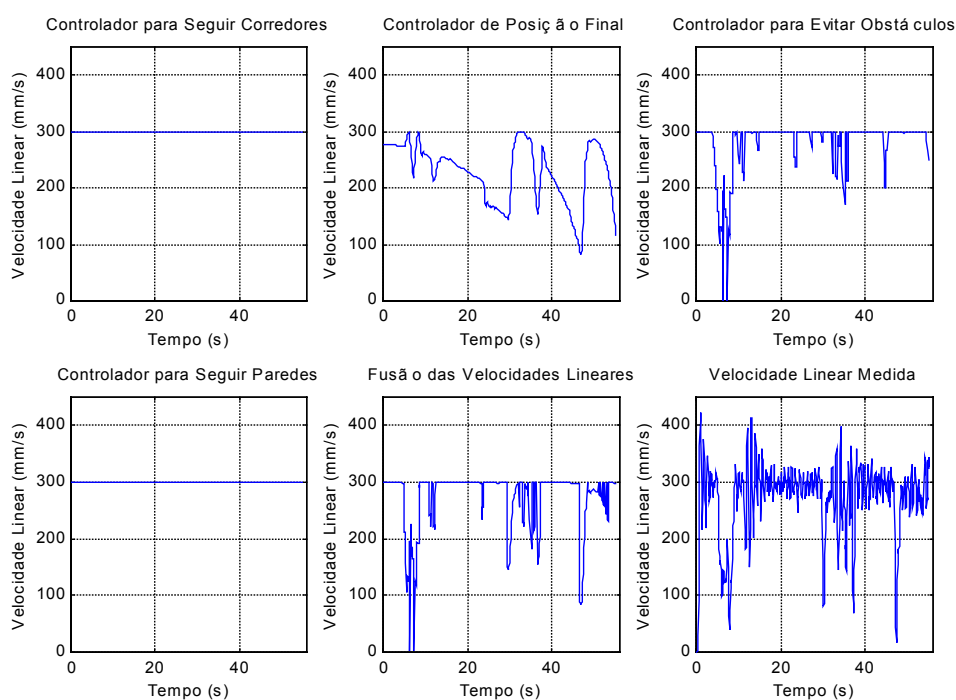


Figura 96 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento #3).

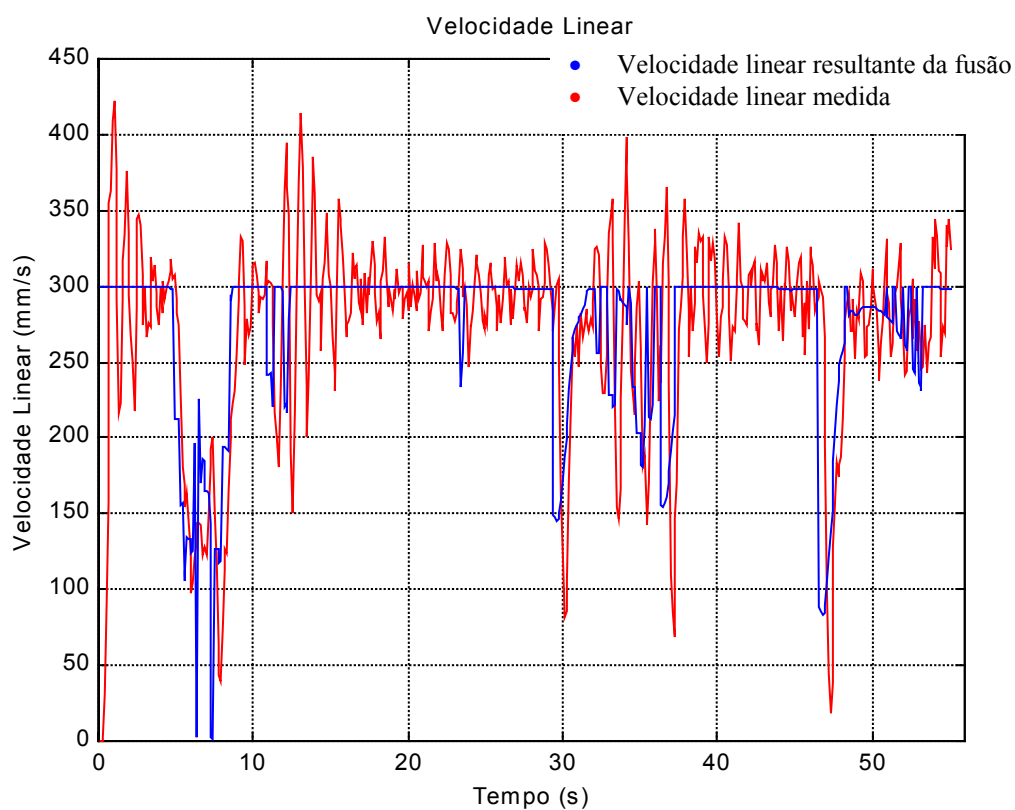


Figura 97 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento #3).

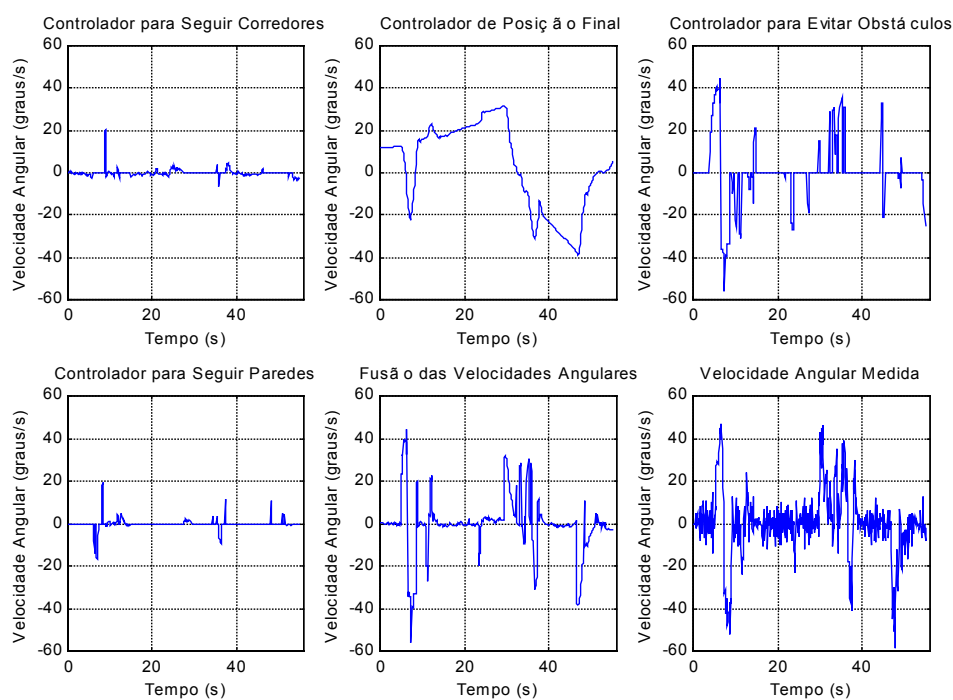


Figura 98 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #3).

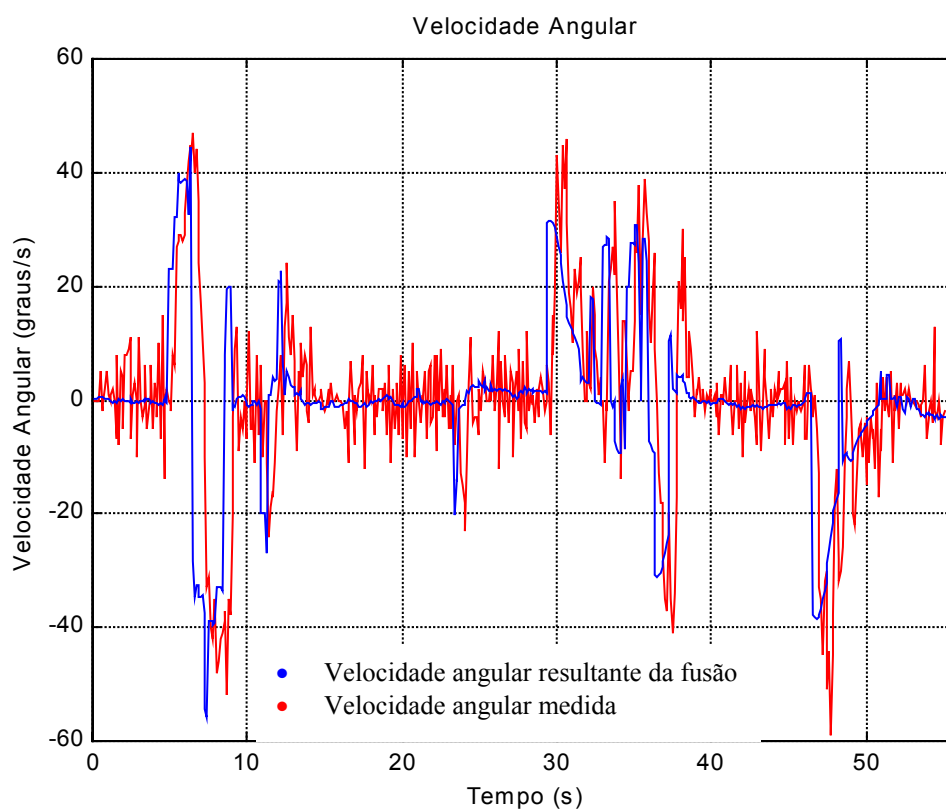


Figura 99 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #3).

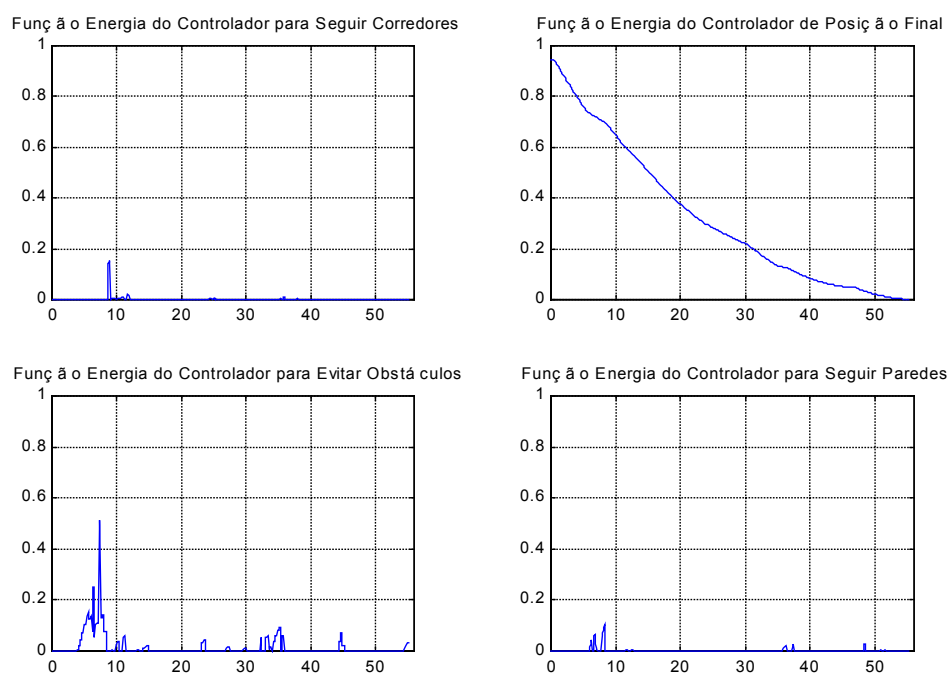


Figura 100 Funções energia associadas aos quatro controladores envolvidos no processo (Experimento #3).

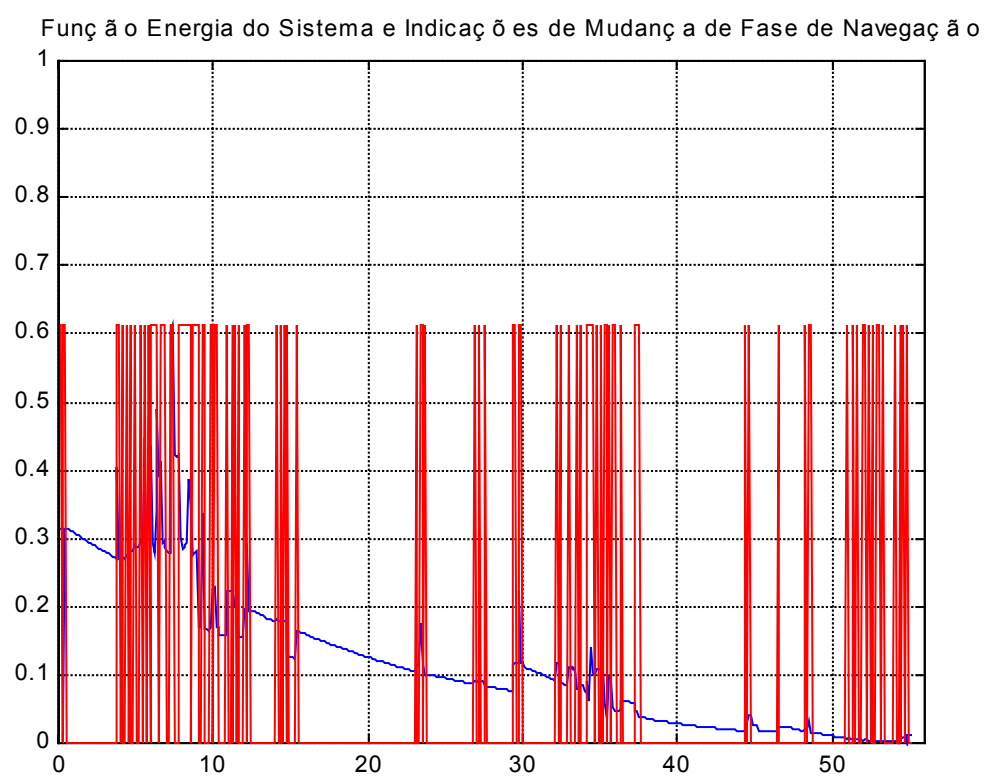


Figura 101 Função energia do sistema e indicações de mudanças de fase de navegação (Experimento #3).

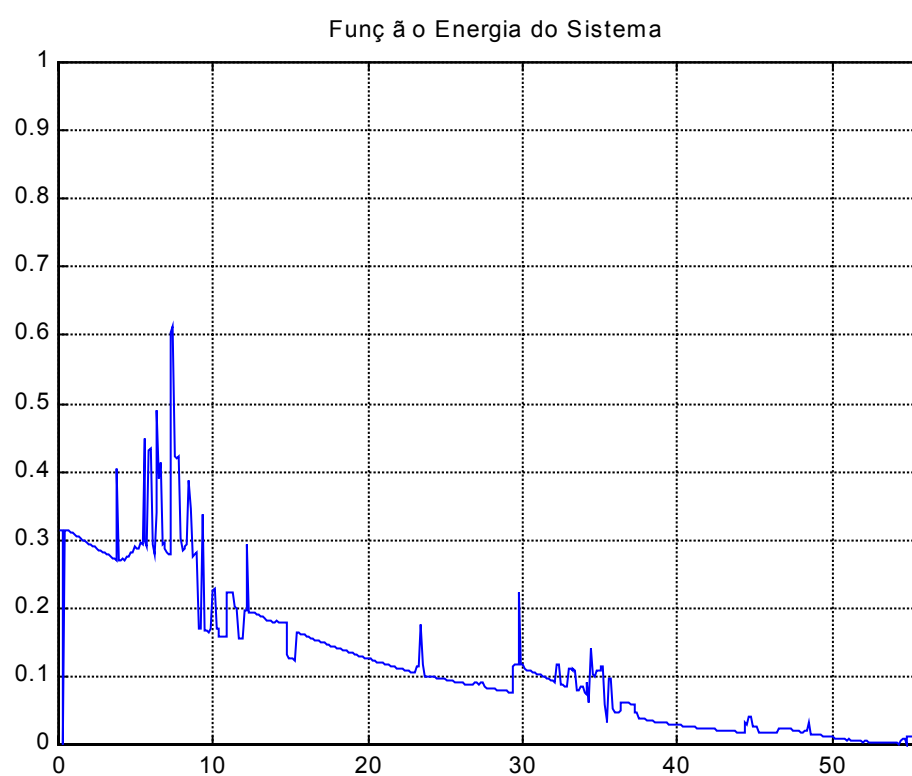


Figura 102 Função energia do sistema (Experimento #3).

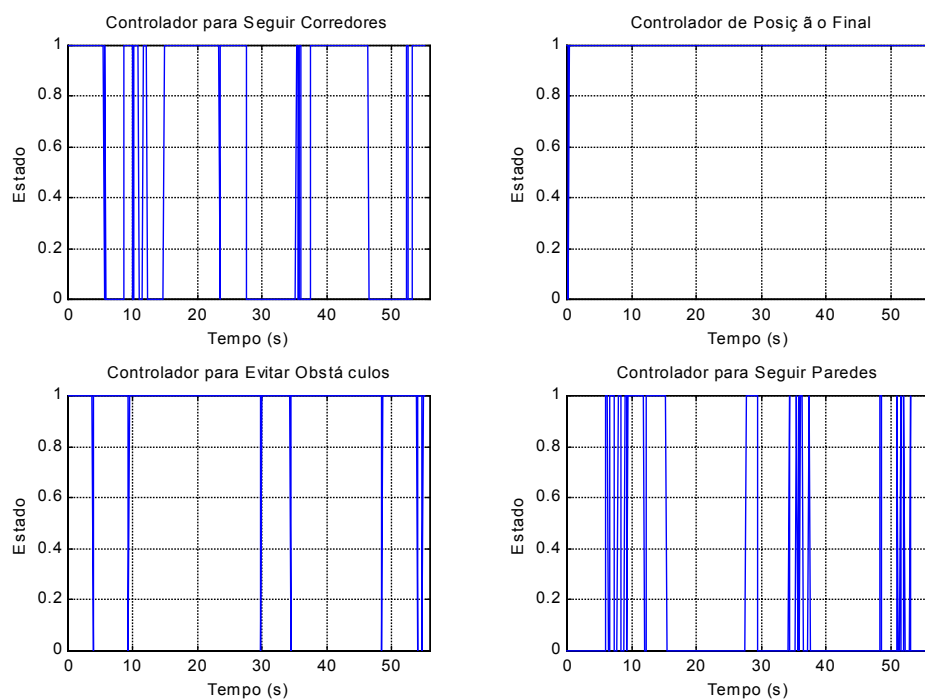


Figura 103 Estados dos controladores (Experimento #3).

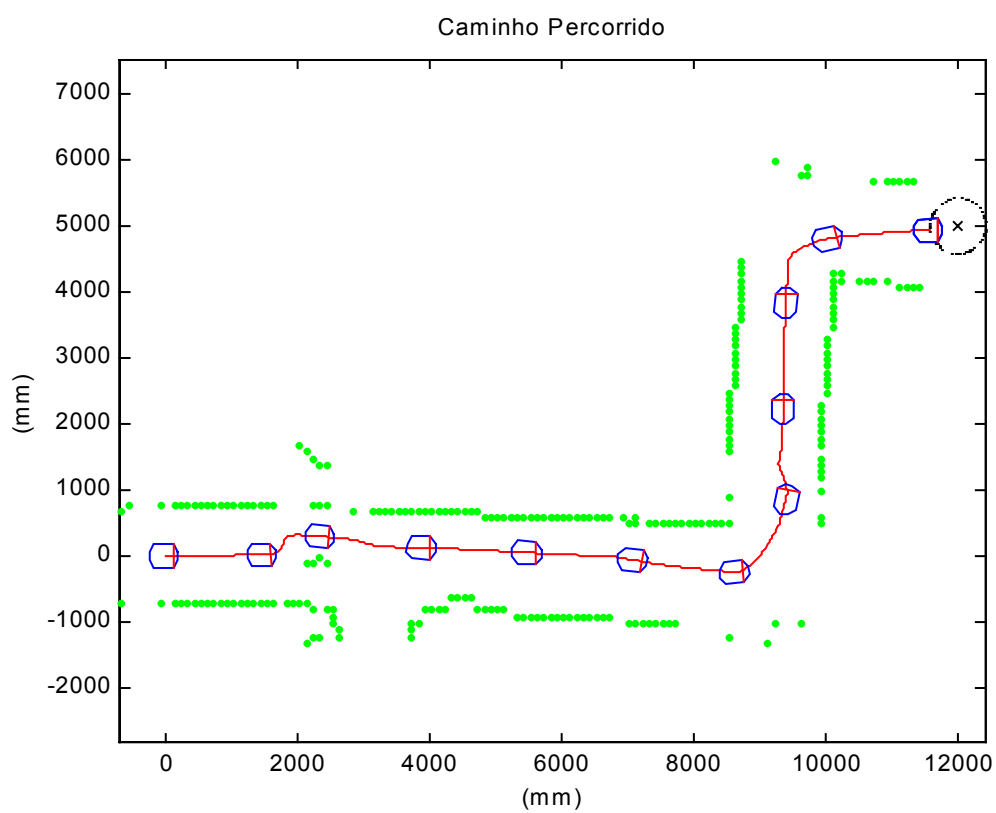


Figura 104 Caminho percorrido pelo robô (Experimento #3).

### 6.2.4 Teste #4

Neste último teste, o objetivo que se deseja alcançar consiste em guiar o robô de um ponto inicial, nas coordenadas [0m, 0m], até o ponto de destino, nas coordenadas [5m, 5,5m], em um ambiente composto por corredores, portas, paredes e obstáculos. O ponto de partida e o ponto de destino não estão no mesmo corredor, e para o robô chegar ao seu objetivo, o caminho mais curto assemelhasse à forma de um “U”. Este teste também visa demonstrar a capacidade que a arquitetura proposta possui de evitar pontos de mínimo locais.

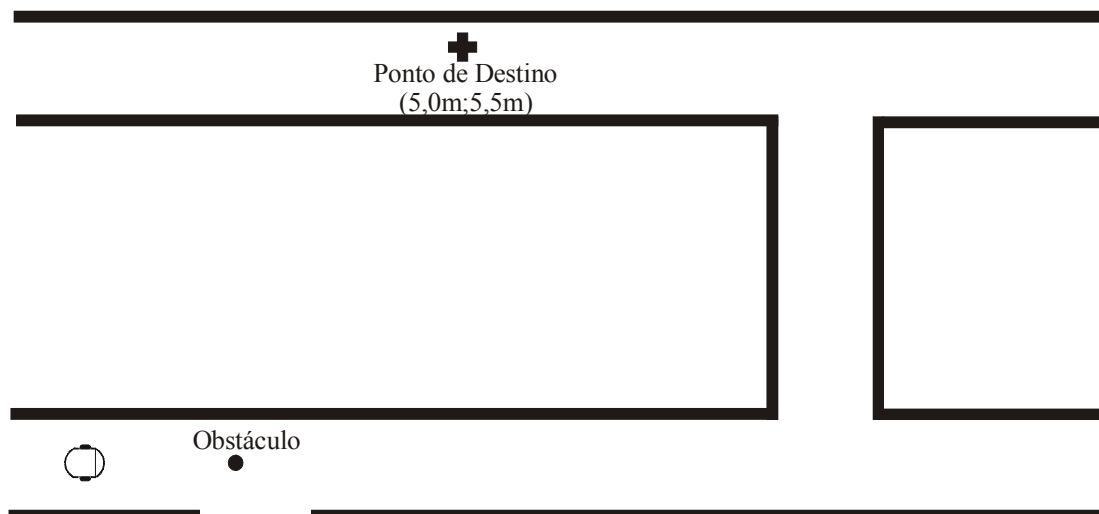


Figura 105 Teste #4

Tabela 10 Índices de Avaliação de Desempenho do Teste #4.

Índice	Simulação	Experimento	Valores Ideais
Segurança	186 mm	142 mm	500 mm
Velocidade Média	272 mm/s	287 mm/s	300 mm/s
Suavidade	1,22°	0,99°	0,31°
Distância Percorrida	17,73 m	18,11 m	17,50 m
Tempo	65,20 s	63,20 s	58,33 s

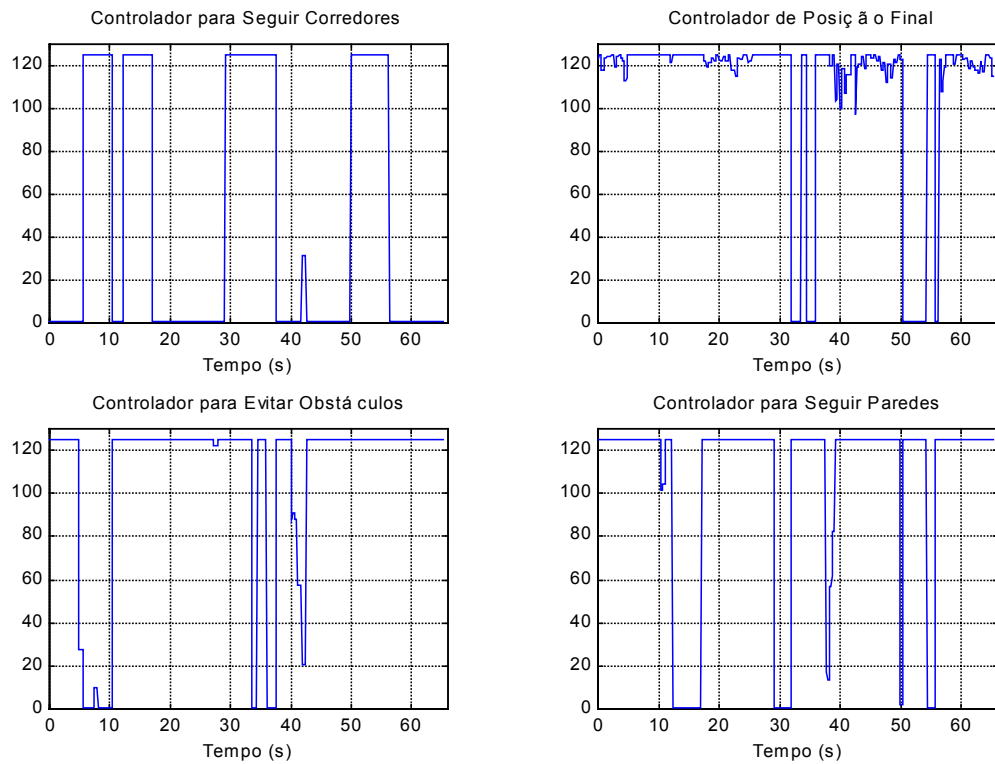


Figura 106 Covariâncias associadas aos controladores (Simulação #4).

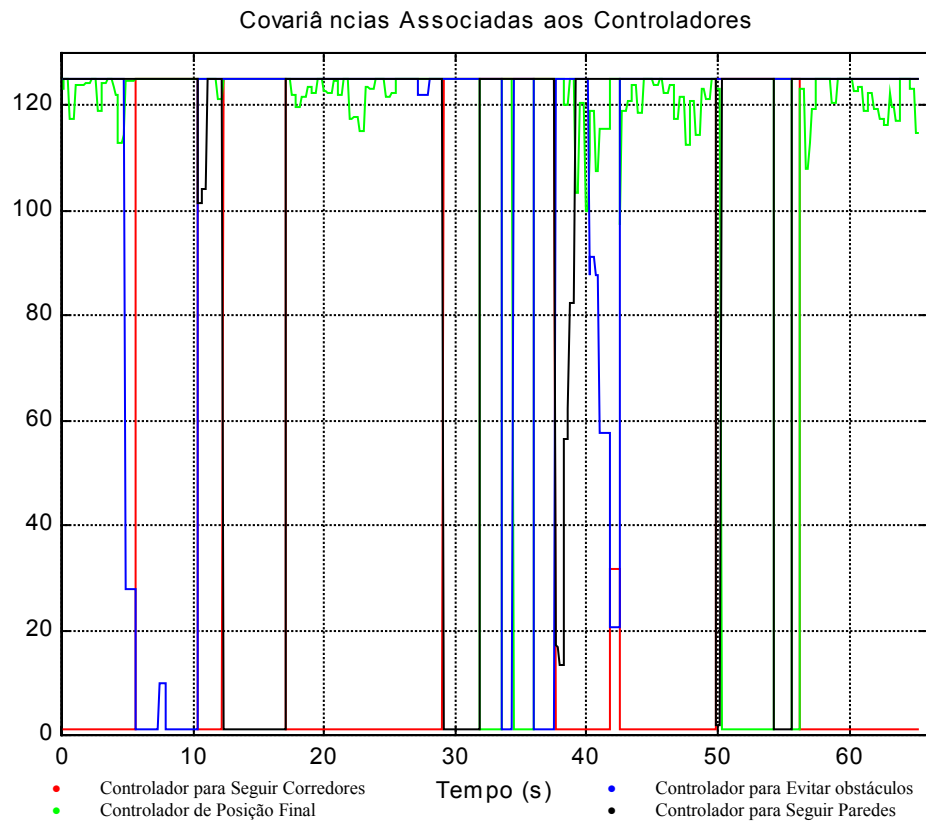


Figura 107 Covariâncias associadas aos controladores sobrepostas (Simulação #4).



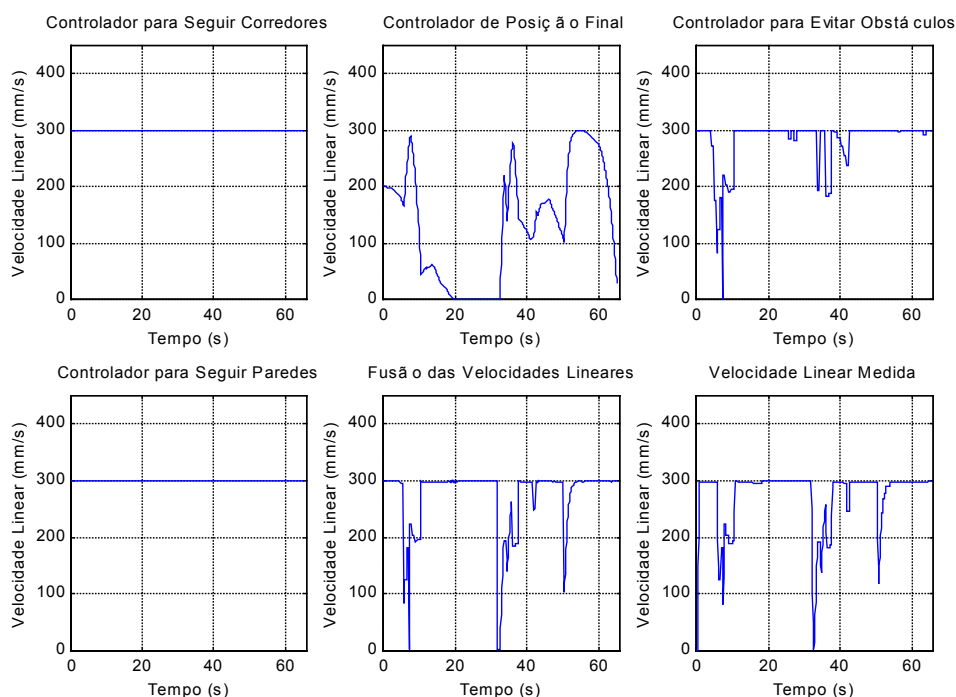


Figura 108 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Simulação #4).

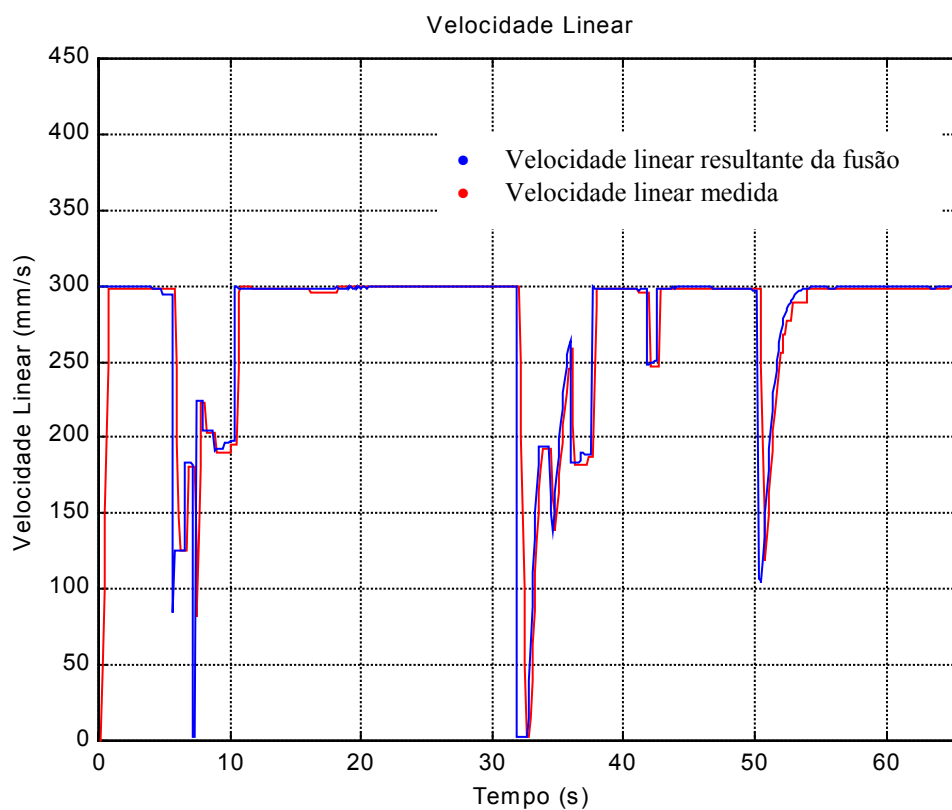


Figura 109 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Simulação #4).

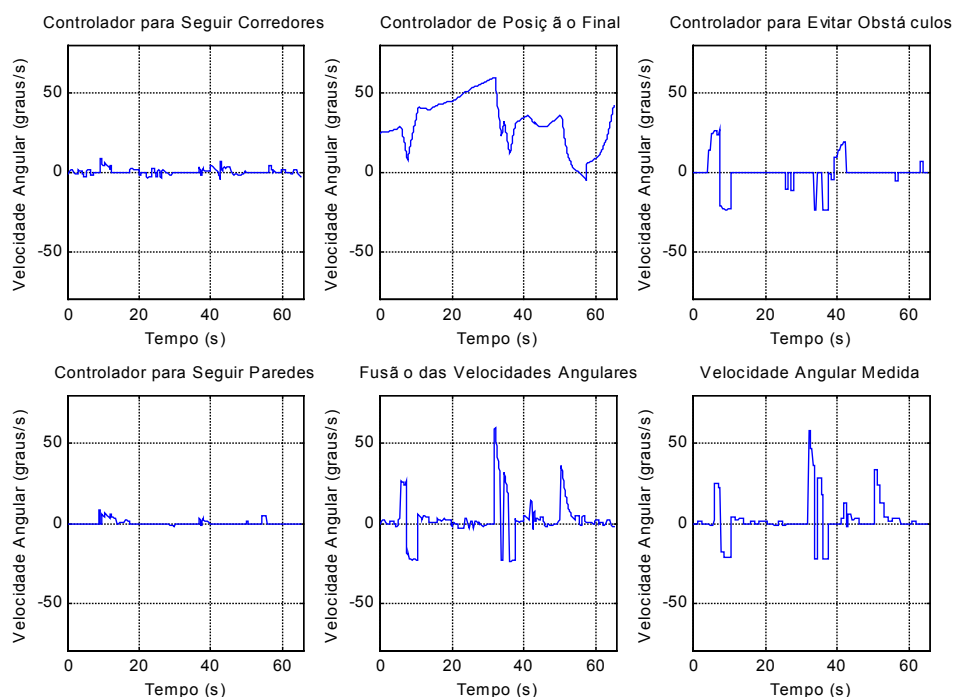


Figura 110 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Simulação #4).

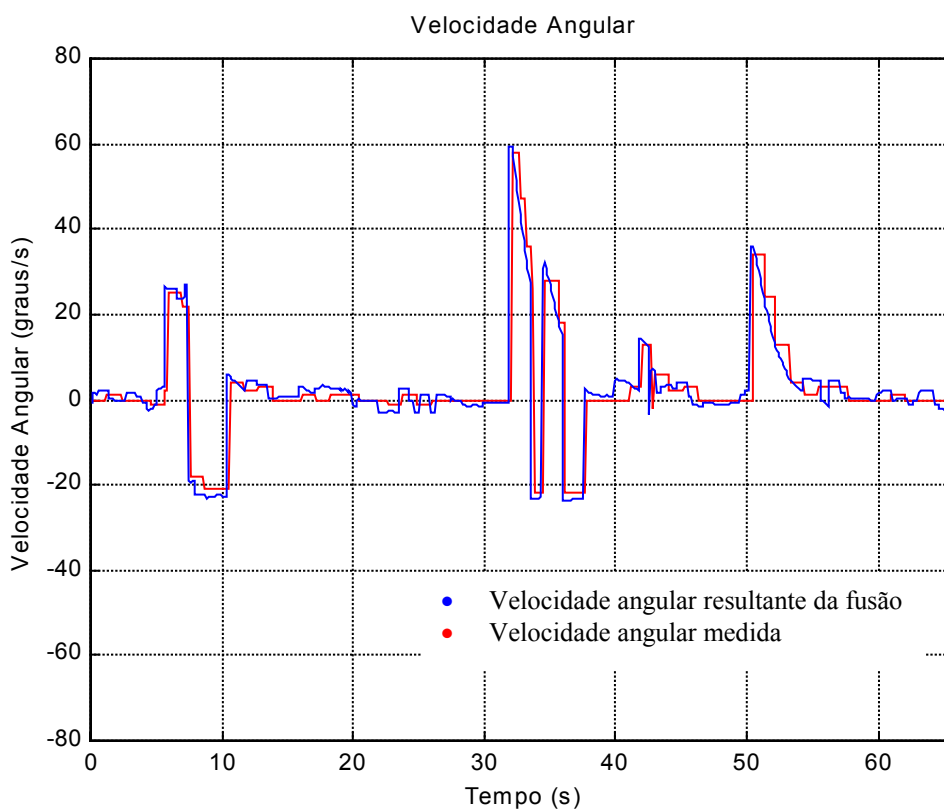


Figura 111 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (simulação #4).

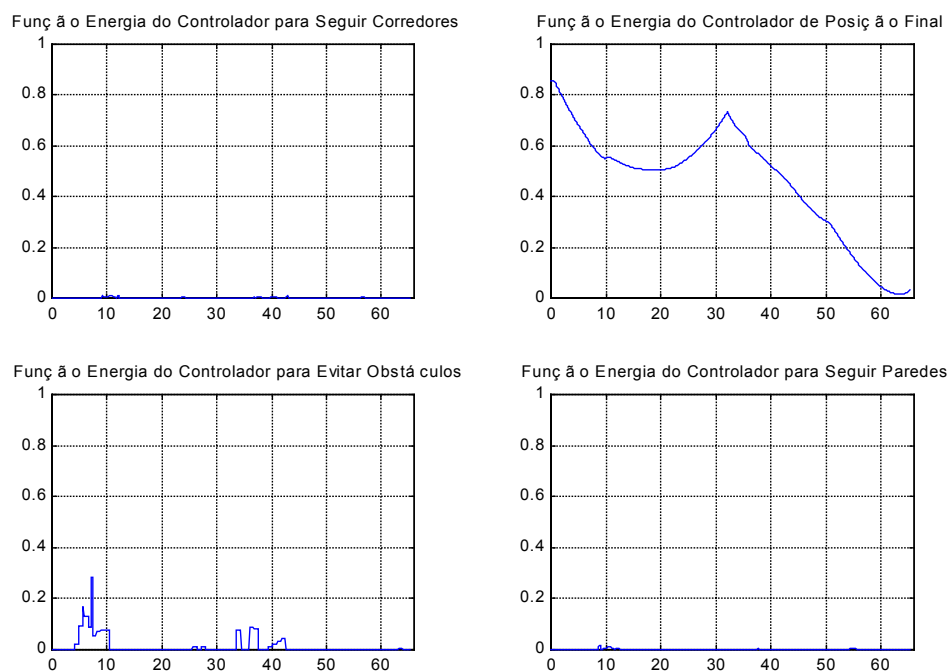


Figura 112 Funções energia associadas aos quatro controladores envolvidos no processo (Simulação #4).

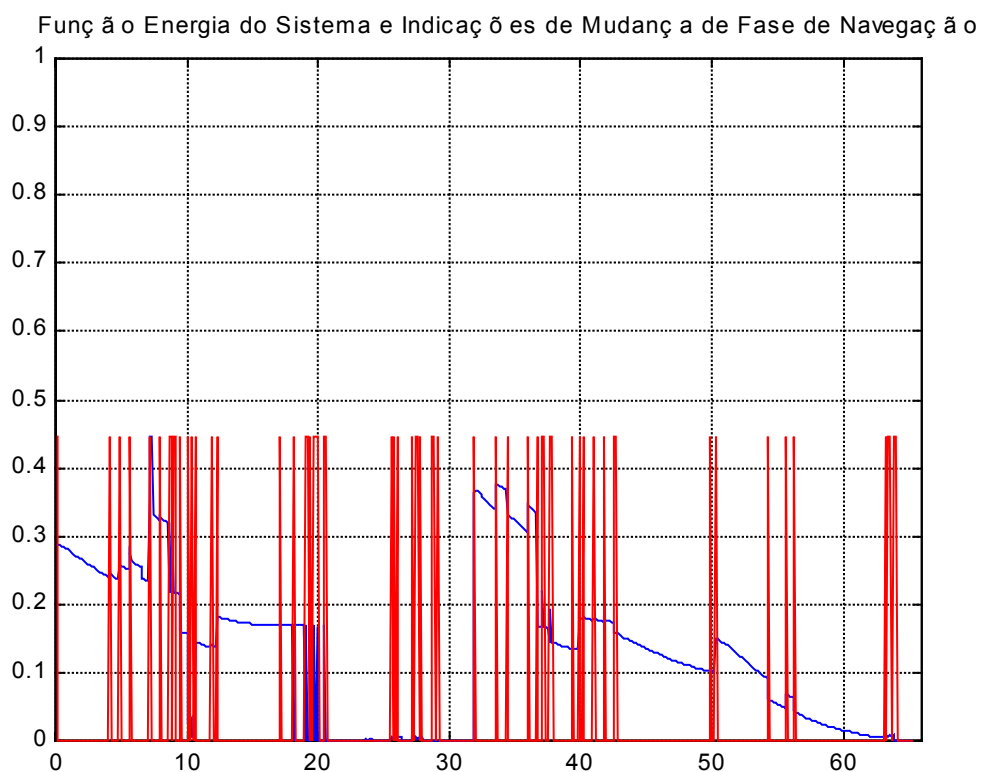


Figura 113 Função energia do sistema e indicações de mudanças de fase de navegação (Simulação #4).

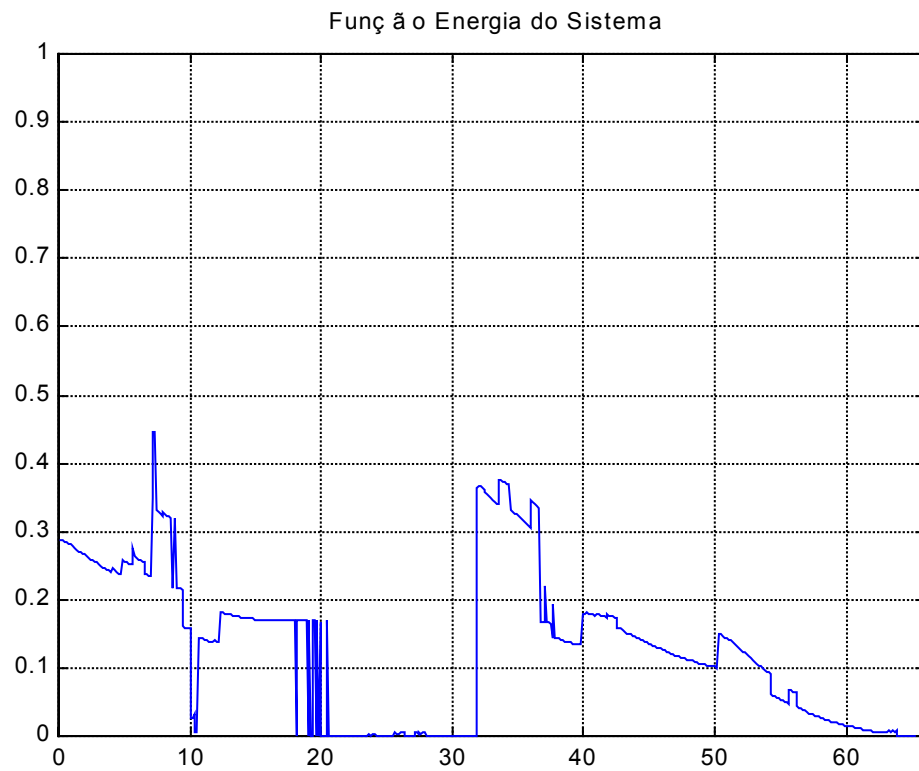


Figura 114 Função energia do sistema (Simulação #4).

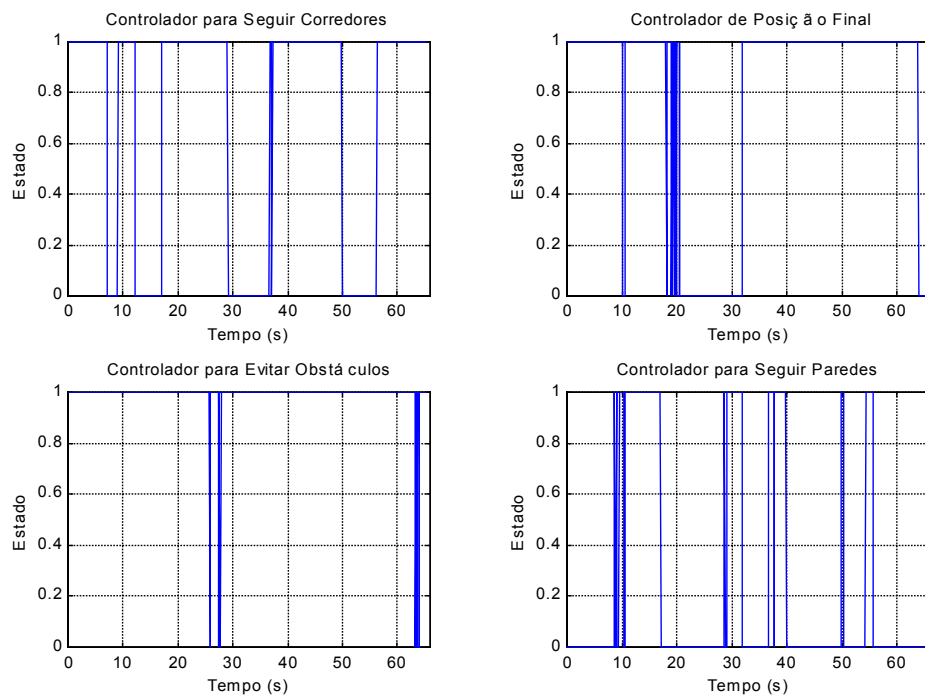


Figura 115 Estados dos controladores (Simulação #4).

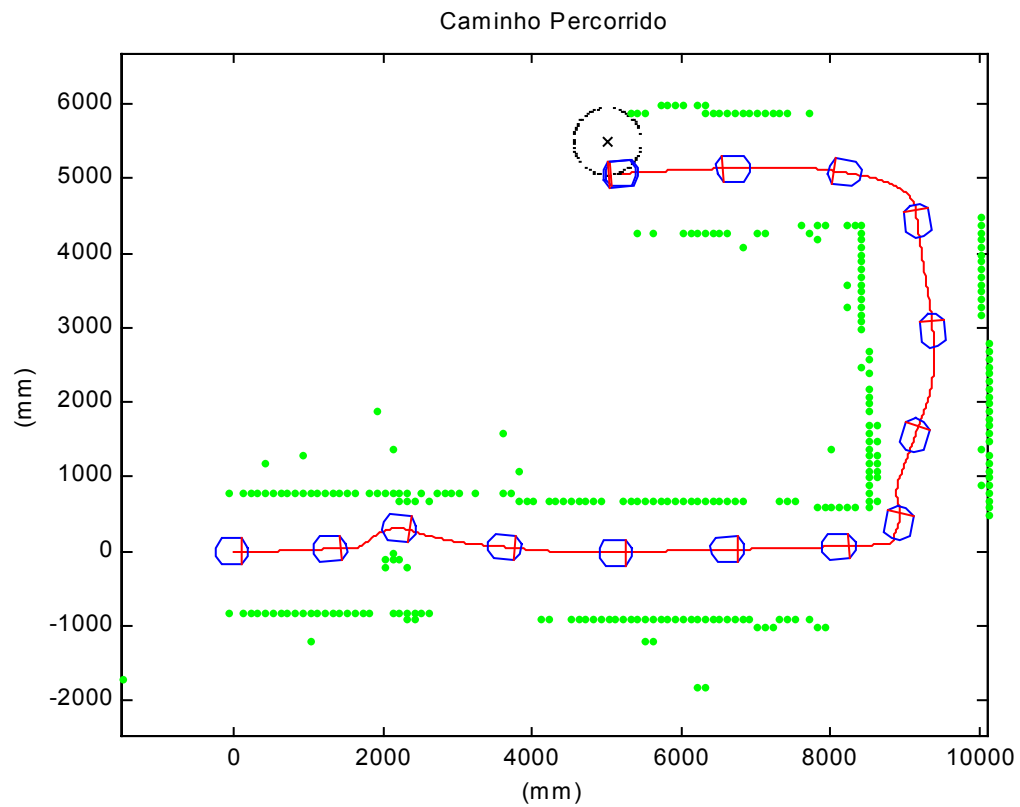


Figura 116 Caminho percorrido pelo robô (Simulação #4).

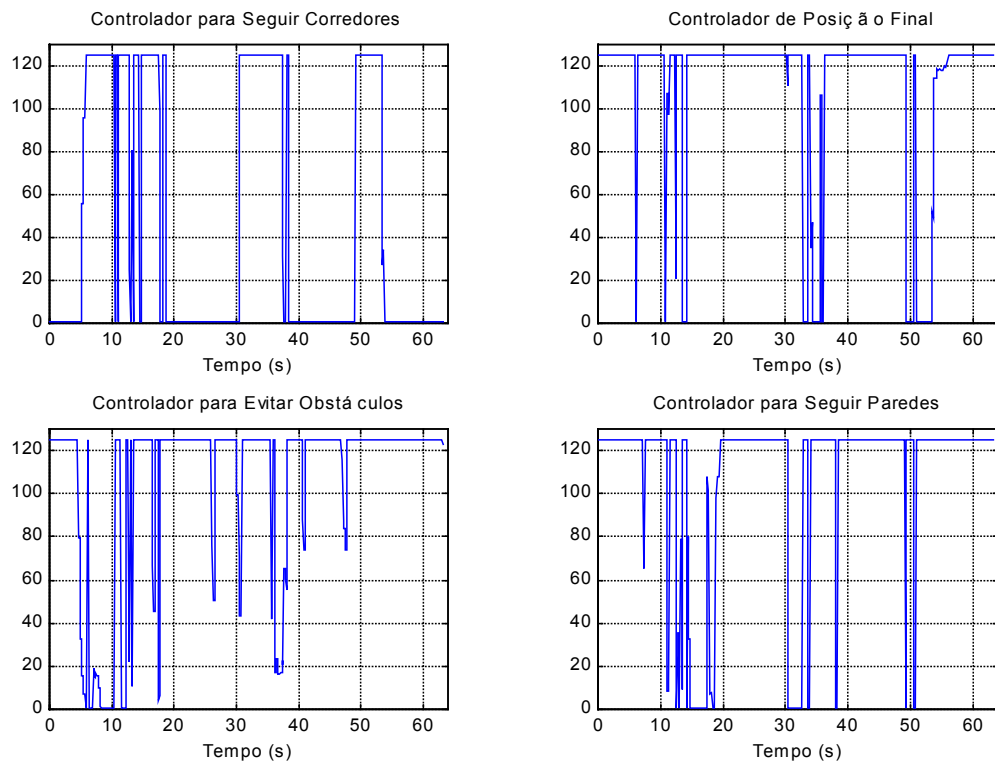


Figura 117 Covariâncias associadas aos controladores (Experimento #4).

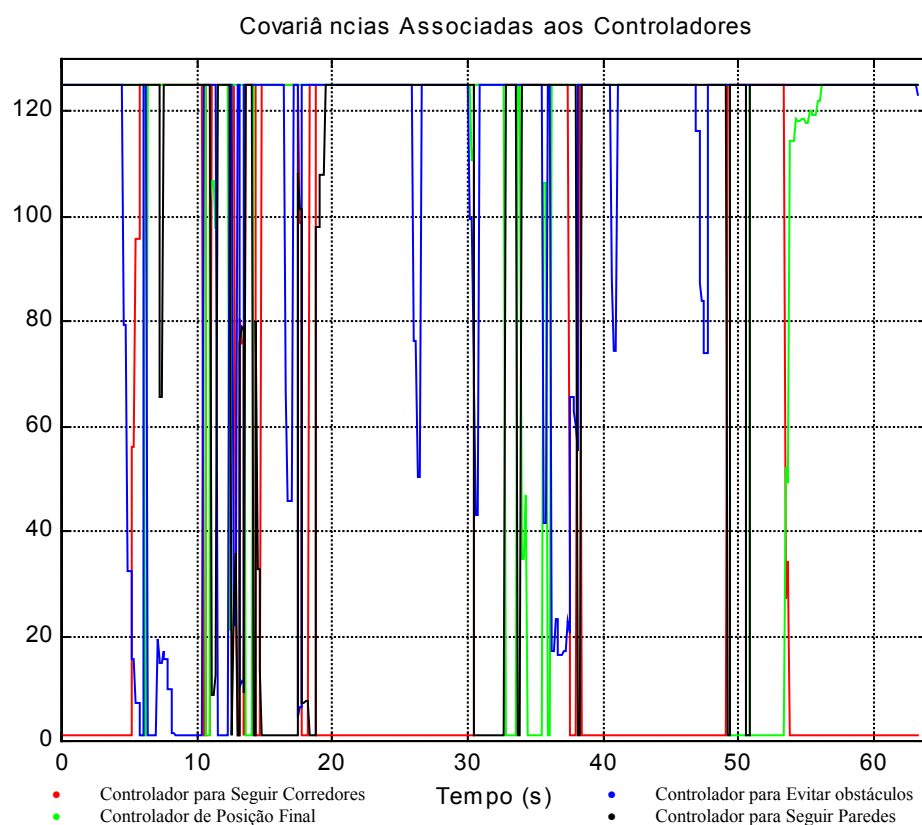


Figura 118 Covariâncias associadas aos controladores sobrepostas (Experimento #4).

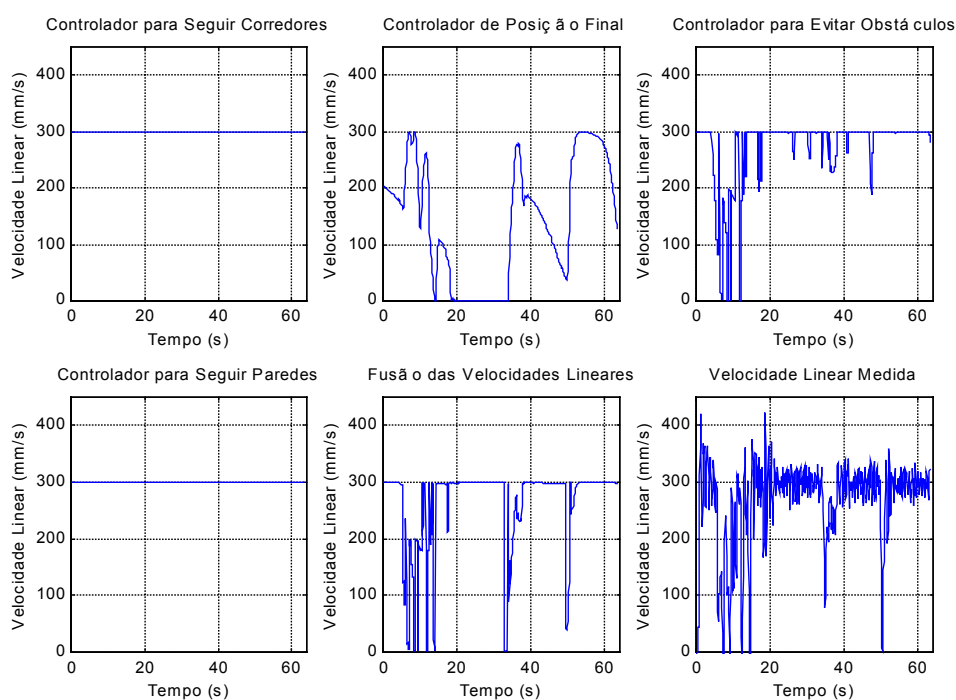


Figura 119 Velocidades lineares produzidas pelos quatro controladores, além da velocidade linear resultante do processo de fusão e da velocidade linear medida pelo robô (Experimento #4).

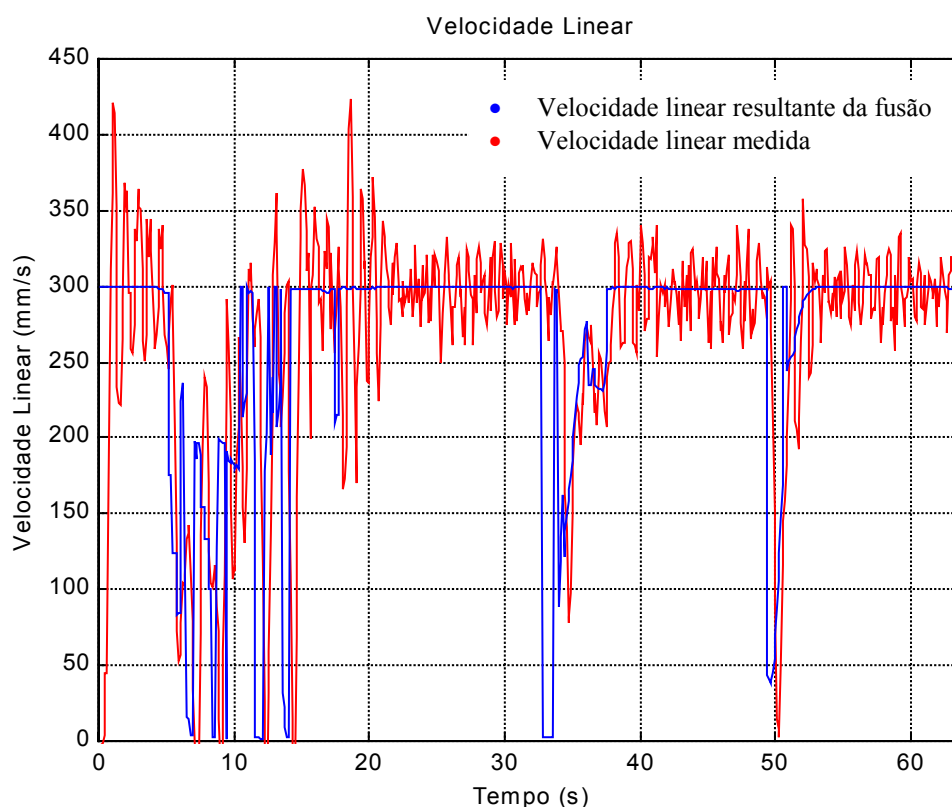


Figura 120 Velocidade linear resultante do processo de fusão e velocidade linear medida pelo robô (Experimento #4).

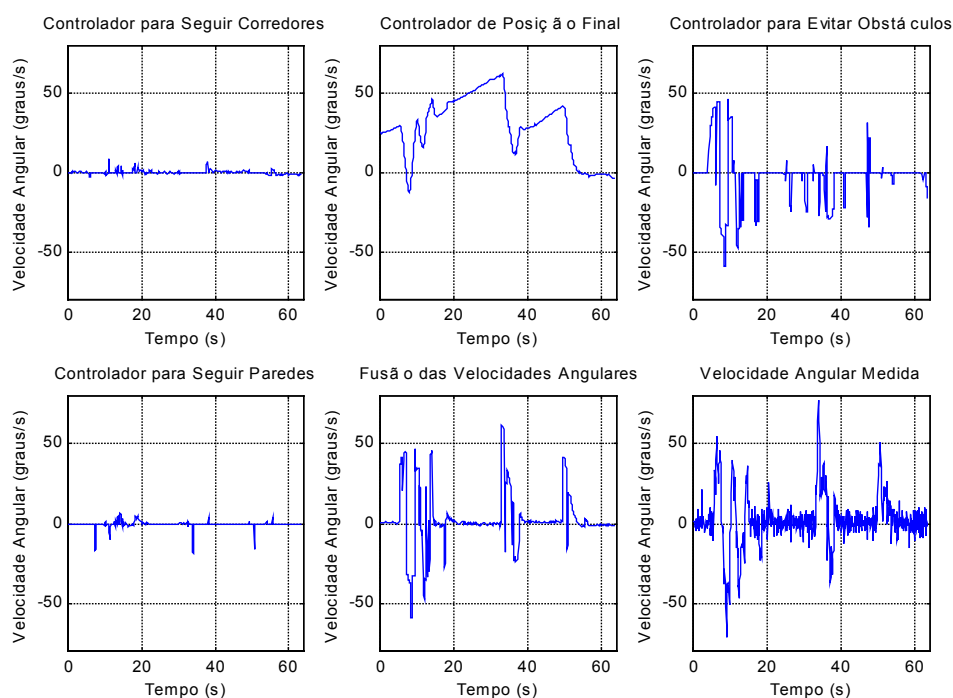


Figura 121 Velocidades angulares produzidas pelos quatro controladores, além da velocidade angular resultante do processo de fusão e da velocidade angular medida pelo robô (Experimento #4).

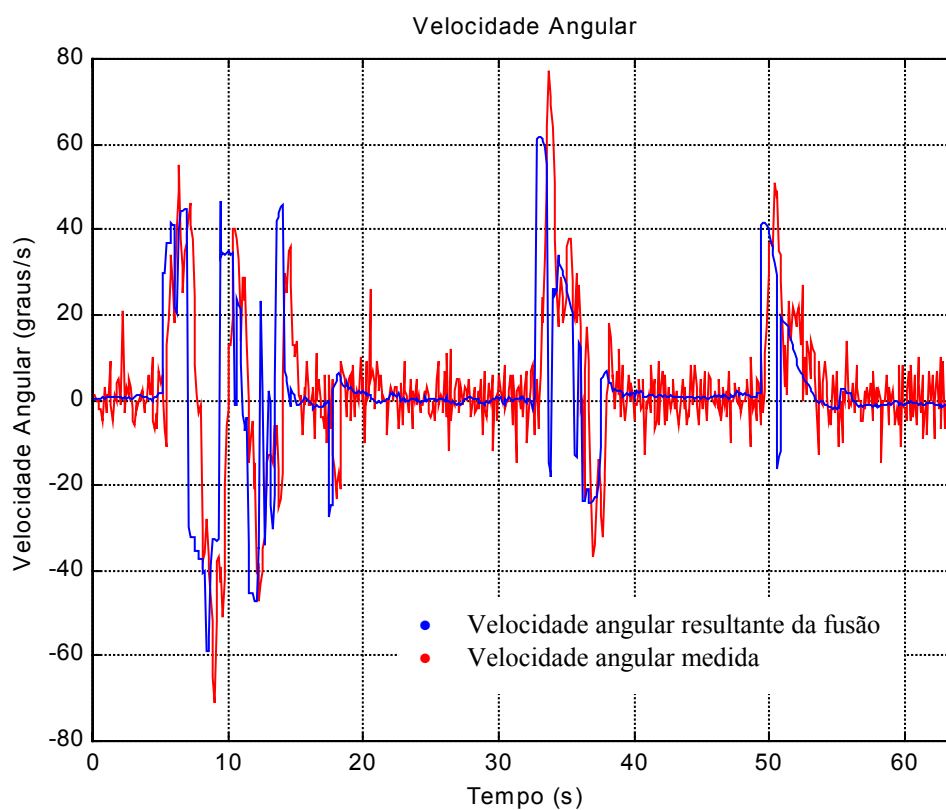


Figura 122 Velocidade angular resultante do processo de fusão e velocidade angular medida pelo robô (Experimento #4).

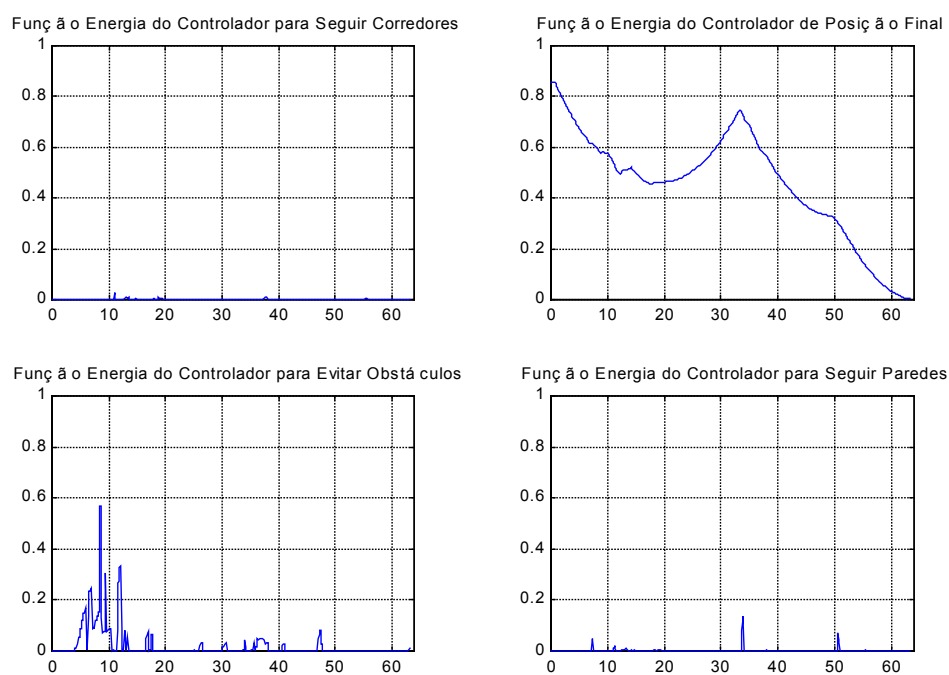


Figura 123 Funções energia associadas aos quatro controladores envolvidos no processo (Experimento #4).



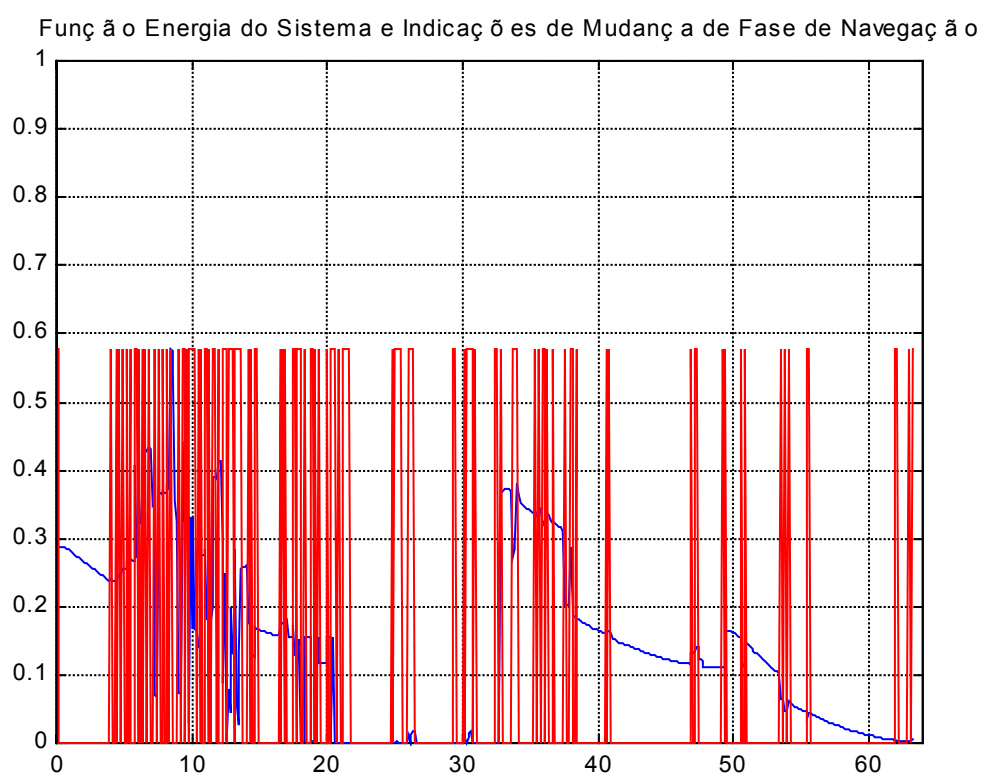


Figura 124 Função energia do sistema e indicações de mudanças de fase de navegação (Experimento #4).

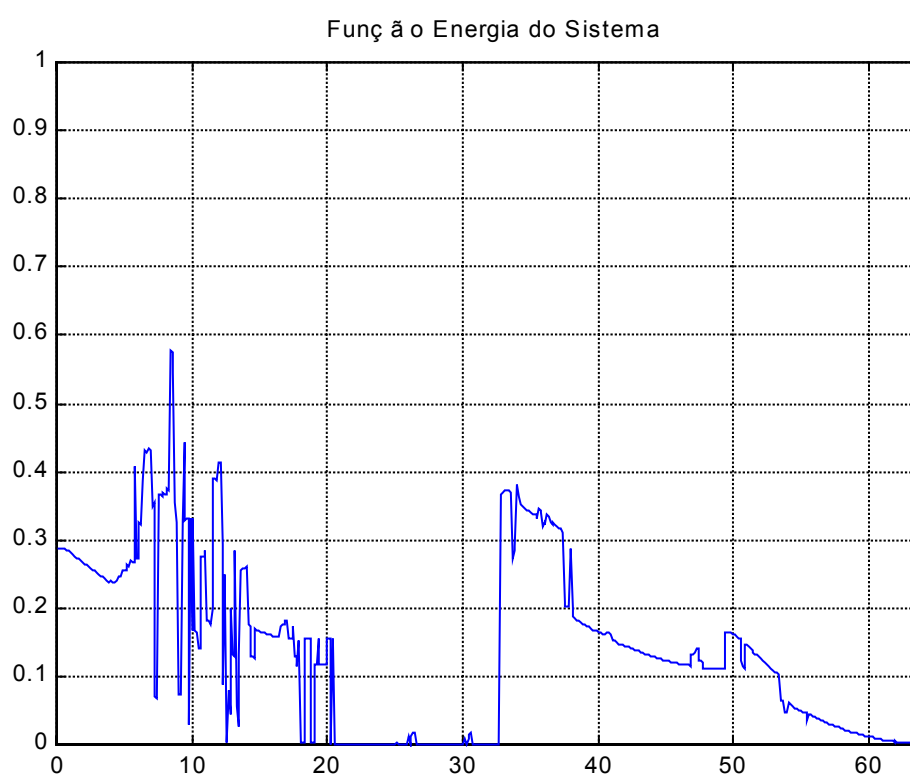


Figura 125 Função energia do sistema (Experimento #4).

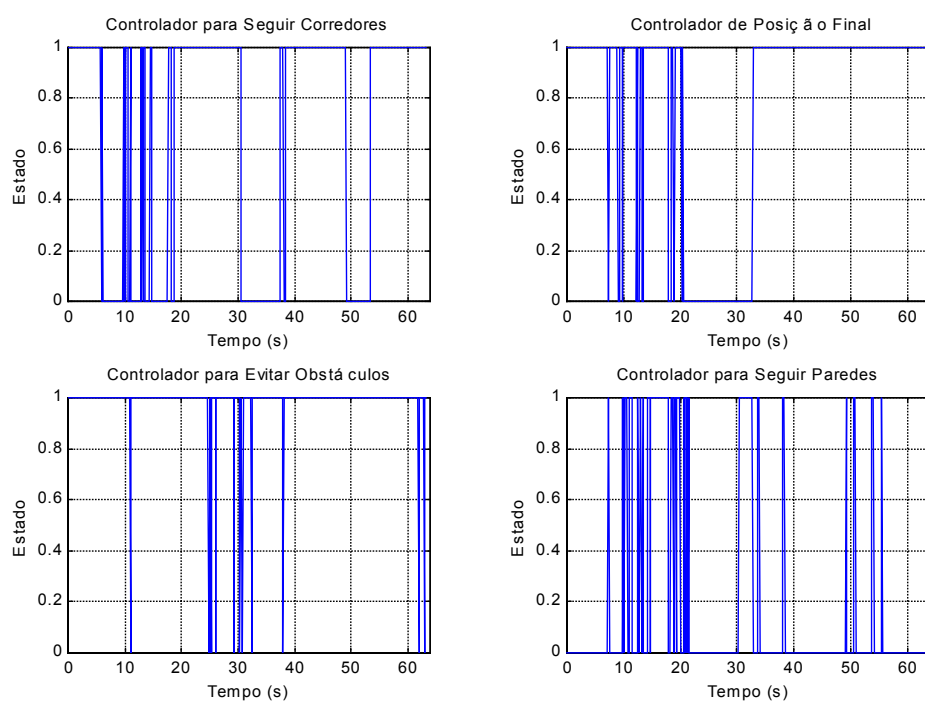


Figura 126 Estados dos controladores (Experimento #4).

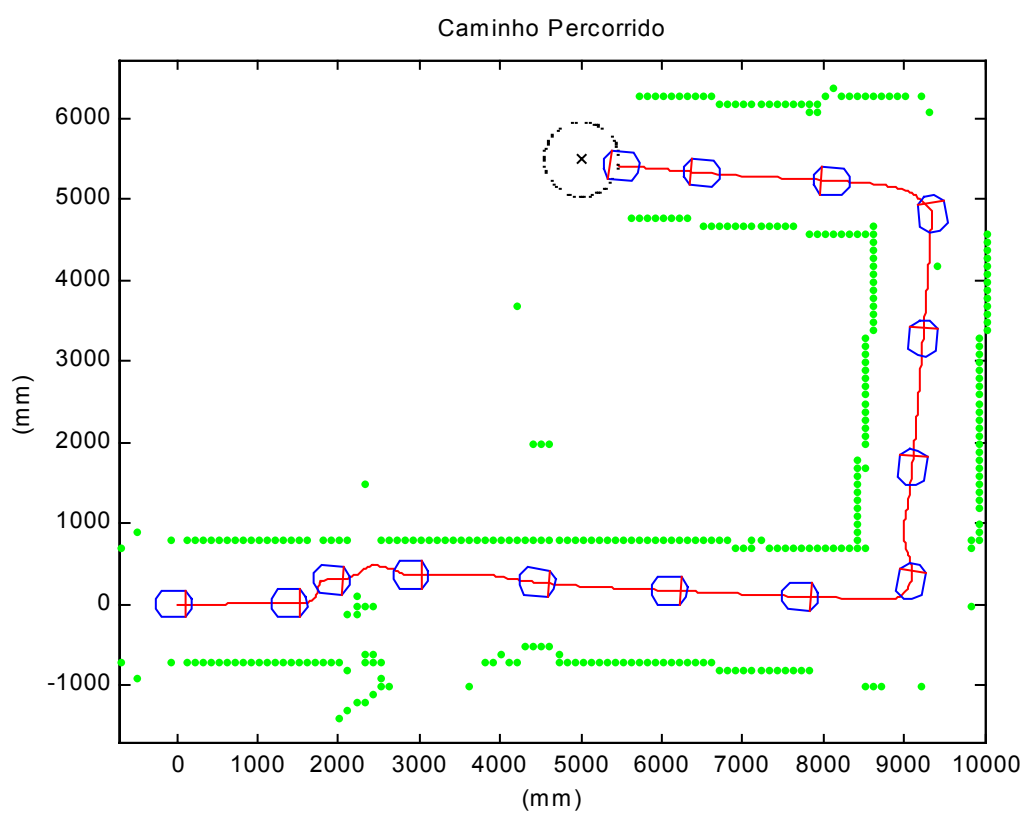


Figura 127 Caminho percorrido pelo robô (Experimento #4).

### 6.3 *Análise dos Resultados*

Como ilustram as simulações e experimentos apresentados neste capítulo, o robô percorre trajetos seguros e executa manobras suaves.

A velocidade linear média é um pouco menor do que os valores ideais, devido ao fato de que a fusão de diferentes sinais de controle faz com que o robô navegue um pouco mais lento.


As simulações e os experimentos também demonstram que a arquitetura de controle proposta, foi capaz de evitar os pontos de mínimo local (um problema clássico) tanto no Teste #2 quanto no Teste #4 e que a conjectura de estabilidade, proposta para o caso em que estão envolvidos controladores com diferentes objetivos de controle, se cumpre.

Se for realizada uma comparação dos gráficos onde se apresentam a velocidade angular medida pelo robô em simulações com os gráficos análogos para o caso de experimentos reais, fica claro que o simulador *Saphira* não leva em consideração o modelo correto da dinâmica do robô. Para os experimentos realizados, se for realizada ainda uma comparação entre os gráficos que apresentam a velocidade angular resultante do processo de fusão e a velocidade angular medida pelo robô, fica clara a necessidade de no futuro levantar o modelo dinâmico do robô, para que a sua dinâmica possa ser compensada, o que permitirá reduzir as oscilações observadas, principalmente na velocidade angular do robô.

Em casos onde o robô tivesse que se aproximar muito de um obstáculo em potencial, como seria o caso do acoplamento a uma estação de recarga, o sistema de controle proposto na Figura 9 falharia, pois o controlador de evitar obstáculos afastaria o robô de seu objetivo final. Neste caso porém, a falha não é da arquitetura de controle proposta, e sim do sistema de controle implementado para exemplificar o seu funcionamento. Para que esta tarefa fosse resolvida com sucesso, o projetista do sistema teria que levá-la em consideração durante o projeto do sistema de controle, o que no caso do sistema apresentado na Figura 9 não ocorreu. Acontece que esta fase de navegação não foi inicialmente prevista, e não existe nenhum controlador para lidar com ela. Seria necessário então incorporar ao sistema um controlador para executar tal tarefa de acoplamento.

## Capítulo 7

### Conclusões



Nesta Tese de Doutorado é apresentada uma nova arquitetura de controle, que pode ser aplicada ao controle de robôs móveis e foi denominada *Fusão de Sinais de Controle*. O método de fusão empregado é o *Filtro de Informação Descentralizado* (FID), também proposto nesta Tese, o qual é utilizado para realizar a fusão dos vários sinais de controle para as velocidades angular e linear provenientes dos diversos controladores. A escolha do FID foi baseada em uma comparação do Filtro de Informação com o Filtro de Kalman, que deixou claras algumas vantagens que o primeiro possui com relação ao segundo (vide Capítulo 2). Esta Tese também propôs o uso de lógica *fuzzy* para definir a covariância associada a cada controlador.

Também é apresentado um exemplo de implementação de um sistema de controle baseado na arquitetura proposta, composto por um controlador para seguir corredores, um controlador para seguir paredes, um controlador para evitar obstáculos e um controlador de Posição Final, que permite ao robô chegar a um ponto destino. Destes, os controladores para seguir paredes e corredores foram originalmente propostos neste trabalho e o controlador para evitar obstáculos foi adaptado a partir do controlador baseado em impedância [24][25][48].

Algumas outras arquiteturas de controle recentemente propostas foram apresentadas resumidamente na introdução desta Tese, e com relação a elas, a arquitetura proposta apresenta duas vantagens principais. A primeira consiste no fato de que a arquitetura de controle proposta utiliza como mecanismo de fusão o FID, que é um estimador ótimo [1][13]. A segunda vantagem consiste no fato de que a fusão é realizada diretamente sobre as saídas dos distintos controladores (ou seja, velocidades lineares e angulares), evitando, portanto, qualquer tipo de pré-processamento dos dados antes do processo de fusão.

Nesta Tese também foram discutidos aspectos de estabilidade associados à nova arquitetura de controle.

É apresentada uma demonstração formal de que a fusão das saídas de diferentes controladores com mesmos objetivos de controle possui uma solução *ultimately bounded*. Desta forma, o sinal de controle resultante da fusão é melhor do que os sinais de controle produzidos por cada controlador separadamente, no sentido de que a variância do erro após a fusão é menor do que a variância do erro de cada controlador envolvido no processo de fusão.

O trabalho também traz considerações a respeito da estabilidade da fusão das saídas de diferentes controladores com diferentes objetivos de controle. Neste caso, uma conjectura de estabilidade é apresentada e validada através de várias simulações e experimentos, sendo vários deles aqui apresentados.

A conclusão baseada em análise matemática e resultados experimentais apresentados nesta Tese de Doutorado é que a arquitetura proposta, baseada na fusão das saídas de diferentes controladores, efetivamente apresenta um “bom comportamento” para uma aplicação como a navegação de um robô móvel em ambientes complexos, como por exemplo, um ambiente composto por corredores, paredes e obstáculos. Isto significa que o robô não perdeu o seu objetivo final, mesmo quando existiam obstáculos em seu caminho ou quando o ambiente o obrigava a desviar-se temporariamente do seu destino final.

A arquitetura de controle proposta é bastante flexível, podendo ser incluído qualquer controlador, desde que as suas variáveis de saída sejam, ou possam ser transformadas em, velocidade linear e/ou velocidade angular. As exigências de sincronismo entre os controladores e de que os controladores conduzam a sistemas de controle estáveis no sentido de Lyapunov podem ser relaxadas quando não seja necessário levar em consideração a questão da estabilidade do sistema. Também não há restrição (exceto as restrições computacionais) quanto ao número de controladores utilizados.

A utilização da arquitetura de controle proposta é adequada para a maioria das aplicações onde vários objetivos de controle, muitas vezes em conflito, necessitam ser considerados ao mesmo tempo.

Como um trabalho futuro, seria interessante realizar um estudo comparativo entre a arquitetura de controle proposta e aquelas mencionadas no Capítulo 1. Tal estudo comparativo deveria ser baseado em um amplo conjunto de experimentos, repetidos várias vezes cada, de modo que fosse possível realizar uma análise estatística dos resultados obtidos por cada arquitetura de controle. Neste sentido, os índices de desempenho mencionados no Capítulo 6 podem ser utilizados para quantificar o desempenho do robô em cada experimento.

Também seria importante realizar um estudo para realizar uma modelagem estatística das falhas e do ruído associado ao processo de medição dos diversos tipos de sensores que normalmente são aplicados em robôs móveis, levando em consideração a variedade de ambientes onde estes robôs podem operar.

Um outro trabalho futuro seria a realização de um estudo comparativo semelhante ao apresentado no Capítulo 2, mas que o sistema utilizado fosse dinâmico.

Conforme ficou claro ao analisar a diferença entre as curvas de velocidade lineares e angulares obtidas em simulações e experimentos apresentadas no Capítulo 6, é importante também encontrar o modelo dinâmico do robô. Com isso, é possível utilizar a lei de dinâmica

inversa [29] para realizar a compensação da dinâmica do robô, o que resultará em um movimento mais suave.


Para demonstrar a aplicabilidade da arquitetura de controle proposta, poderia ser implementado um sistema de controle baseado nela em um robô móvel com recursos computacionais mais modestos do que os que estão disponíveis no PIONEER 2DX.

Também de grande importância seria a implementação de um sistema de controle baseado na arquitetura de controle proposta, no qual além dos controladores que aparecem na Figura 9 fossem utilizados controladores homogêneos [38], ou seja, controladores com o mesmo objetivo de controle. Isto poderia ser realizado, por exemplo, acrescentando ao sistema de controle apresentado na Figura 9 um controlador para seguir corredores baseado em fluxo ótico.

Como a arquitetura de controle proposta é adequada para casos onde vários objetivos de controle, que podem ser contraditórios, devem ser considerados ao mesmo tempo, ela deve ser adequada para aplicações que envolvam o trabalho cooperativo de dois ou mais robôs móveis. Portanto, um trabalho futuro interessante seria o desenvolvimento de um sistema de controle baseado na arquitetura proposta, para coordenar a cooperação entre dois ou três robôs móveis.

Para finalizar, é importante mencionar que as publicações [13][14]**Erro! A origem da referência não foi encontrada.**[16][17][18] [19][20], foram resultantes do trabalho desenvolvido nesta Tese de Doutorado.

## Bibliografia



Este capítulo apresenta a bibliografia referenciada ao longo de todo o texto e a bibliografia consultada, que também contribuiu bastante com informações importantes para a realização deste trabalho.

### ***Bibliografia Referenciada***

- [1] A. G. O. Mutambara, *Decentralized Estimation and Control for Multi-sensor Systems*. EUA: CRC Press, 1998.
- [2] A. Gelb, J. F. Kasper Jr., R. A. Nash Jr., C. F. Price e A. A. Sutherland Jr., *Applied Optimal Estimation*. 15<sup>a</sup> edição, England: The MIT Press, 1999.
- [3] A. Saffiotti, “Fuzzy Logic in Autonomous Robotics: Behavior Coordination”, *Proceedings of the 6<sup>th</sup> International Conference on Fuzzy Systems*, Barcelona, Espanha, 1997, pp. 573-578.
- [4] A. Saffiotti, “Some Notes on the Integration of Planning and Reactivity in Autonomous Mobile Robots”, *AAAI Spring Symposium on Foundations of Planning*, Stanford, CA, USA, 1993, pp. 122-126.
- [5] A. Saffiotti, “The Uses of Fuzzy Logic in Autonomous Robot Navigation: A Catalogue Raisonné”, *Technical Report TR/IRIDIA/97-6*, 1997.

- [6] A. Saffiotti, E. H. Ruspini e K. Konolige, "Robust Execution of Robot Plans Using Fuzzy Logic", *Fuzzy Logic in Artificial intelligence – IJCAI'93 Workshop, LNAI 847*, Springer-Verlag, Berlim, 1994, pp. 24-37.
- [7] A. Saffiotti, K. Konolige e E. Ruspini. "A multivaluated logic approach to integrating planning and control," *Artificial Intelligence*, 76, pp. 481-526, 1995.
- [8] B. Dev e F. G. Kröse, "Navigation of a mobile robot on the temporal development of the optic flow," *Proceedings of the IEEE/RSJ/GI International Conference. on Intelligent Robots and Systems IROS'97*, Grenoble, 1997, pp. 558-563.
- [9] B. P. Gerkey, M. J. Mataric e G. S. Sukhatme, "Exploiting Physical Dynamics for Concurrent Control of a Mobile Robot", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002)*, Washington, DC, USA, 2002.
- [10] C. M. Soria, *Integración Sensorial Aplicada a un Robot Móvil*, Tese de Mestrado, Universidad Nacional de San Juan – Argentina, 2000.
- [11] E. Bicho, *The dynamic approach to behavior-based robotics*. Tese de Doutorado, Universidade do Minho, Portugal, 1999.
- [12] E. O. Freire, Desenvolvimento de um Sistema de Sensoriamento Ultra-Sônico para um Robô Móvel com Controle Baseado em Agentes. Dissertação de Mestrado, Universidade Federal do Espírito Santo, Brasil, 1997.
- [13] E. O. Freire, R. Carelli, V. Mut, C. Soria, T. F. Bastos-Filho e M. Sarcinelli-Filho, "Mobile robot navigation based on the fusion of control signals from different controllers". *Proceedings of the 2001 European Control Conference*, Porto, Portugal, 2001, pp. 1828-1833.
- [14] E. O. Freire, R. Carelli, V. Mut, H. Secchi, T. F. Bastos-Filho e M. Sarcinelli-Filho, "On the Control of a Mobile Robot: A Comparative Study Regarding a New Approach and Two Well-Known Approaches," *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001)*, 2001, Orlando-Florida. 2001.
- [15] E. O. Freire, T. F. Bastos-Filho, M. Sarcinelli-Filho e R. Carelli, "Mobile Robot Control Architecture via Control Output Fusion: Stability Issues," *10<sup>th</sup> Mediterranean Conference on Control and Automation – MED'2002*, Lisboa, Portugal, 2002.
- [16] E. O. Freire, T. F. Bastos-Filho, M. Sarcinelli-Filho e R. Carelli, "Arquitetura de Controle de Robôs Móveis via Fusão das Saídas dos Controladores: Aspectos de Estabilidade", *Anais do Congresso Brasileiro de Automática – XIV CBA*, Natal, Brasil, 2002.



- [17] E. O. Freire, T. F. Bastos-Filho, M. Sarcinelli-Filho e R. Carelli, "A Control Architecture for Mobile Robots Using Fusion of the Output of Distinct Couplers," *Proceedins of the 17th IEEE International Symposium on Intelligent Control*, 2002.
- [18] E. O. Freire, T. F. Bastos-Filho, M. Sarcinelli-Filho, R. Carelli e O. Nasisi, "Uma Nova Arquitetura de Controle para Navegação de Robôs Móveis: Fusão das Saídas de Distintos Controladores," *Simpósio Brasileiro de Automação Inteligente – 7o SBAI*, Caxias do Sul, 2001.
- [19] E. O. Freire, T. F. Bastos-Filho, M. Sarcinelli-Filho, R. Carelli e O. Nasisi, "Navegación de Robots Móviles Basada en Fusión de Salidas de Distintos Controladores," *IX Reunión de Trabajo en Procesamiento de la Información y Control, IX RPIC*, Santa Fe – Argentina, 2001.
- [20] E. O. Freire, T. F. Bastos-Filho, M. Sarcinelli-Filho, R. Carelli e O. Nasisi, "A new mobile robot control architecture via control output fusion," *Proceedings of the XV IFAC World Congress*, Barcelona, Espanha, 2002.
- [21] G. Schöner e M. Dose, "A Dynamics System Approach to Task Level System Integration Used to Plan and Control Autonomous Vehicle Motion," *Robotics and Automation Systems*, 10:253-267, 1992.
- [22] H. A. Simon, *The New Science of Management Decision*. EUA: Harper and Row, 1960.
- [23] H. K. Khalil, *Non-linear systems*, 2<sup>a</sup> edição, USA: Prentice-Hall, 1996.
- [24] H. Secchi, *Control de Vehículos Autoguiados con Realimentación Sensorial*. Dissertação de Mestrado – Universidad Nacional de San Juan – Argentina, 1999.
- [25] H. Secchi, R. Carelli e V. Mut. "Discrete Stable Control of Mobile Robots with Obstacles Avoidance," *International Conference on Advanced Robotics, ICAR'01*, Budapeste, Hungria, 2001.
- [26] J. Borenstein e Y. Koren, "The vector field histogram - Fast obstacle avoidance for mobile robots". *IEEE Transactions on Robotics and Automation*, Vol. 7, No 3, pp. 278-288, 1991.
- [27] J. C. Latombe, *Robot motion planning*. Boston, USA: Kluwer Academic Publishers, 1991.
- [28] J. J. Anaya, C. Fritsch, A. Ruiz, e L. G. Ullate. "A High Resolution Object Recognition Ultrasonic System," *Sensors and Actuators*, Vol. 4, No 37-38, pp. 644-650, 1993.
- [29] J. J. E. Slotine, *Applied Nonlinear Control*. EUA: Prentice-Hall.

- [30] J. K. Rosenblatt, "Utility Fusion: Map-Based Planning in a Behavior-Based System", *Field and Service Robotics, Springer-Verlag*, 1998.
- [31] J. M. M. Abreu, Análisis de senñales ultrasónicas para la definición de entornos en el campo de la robótica. Tese de Doutorado, Universidad Complutense de Madrid, Espanha, 1990.
- [32] J. Rosenblatt, *DAMN: A distributed architecture for mobile navigation*. Tese de Doutorado, Carnegie Melon University, EUA, 1997.
- [33] J. Santos-Victor, G. Sandini, F. Curotto e S. Garibaldi, "Divergent stereo in autonomous navigation: from bees to robots". *International Journal of Computer Vision*, 14-159-177, 1995.
- [34] J. Z. Sasiadek, P. Hartana, "Odometry and sonar data fusion for mobile robot navigation," *6th. IFAC Symposium on Robot Control, SYROCO'00*. Vienna, Austria, 2000. Preprints.
- [35] M. Vidyasagar, *Nonlinear Systems Analysis*. 2ª edição, Prentice-Hall International Editions, 1993.
- [36] P. Althaus, H. I. Christensen e F. Hoffmann, "Using the Dynamical System Approach to Navigate in Realistic Real-World Environments", *Proceedings of the 2001 International Conference on Intelligent Robots and Systems (IROS'2001)*, 2001.
- [37] P. Pirjanian, "Behavior Coordination Mechanisms – State-of-the-art", Technical Report IRIS-99-375, Institute of Robotics and Intelligent Systems, School of Engineering, University of Southern California, EUA, 1999.
- [38] P. Pirjanian, Multiple Objective Action Selection & Behavior Fusion Using Voting, Tese de Doutorado, Aalborg University, Dinamarca, 1998.
- [39] P. Pirjanian, "Multiple objective behavior-based control," *Robotics and Autonomous Systems*, 31, pp. 53-60, 2000.
- [40] P. Pirjanian, H. I. Christensen e J. A. Fayman, "Application of Voting to Fusion of Purposive Modules: An Experimental Investigation," *Robotics & Autonomous Systems*, Vol. 23, No. 4, Julho de 1998. pp 253-266.
- [41] POLAROID Corporation. Ultrasonic ranging system / technical assistance.
- [42] R. A. Brooks, "Achieving artificial intelligence through building robots", MIT A. I. Memo 899, Massachusetts, EUA, 1986.
- [43] R. Babuska, *Fuzzy modelling for control*. EUA: Kluwer Academic Publishers, 1998.

- [44] R. Benporad, M. Di Marco e A. Tesi, "Wall-following controllers for sonar-based mobile robots," *Proceedings of 36th. IEEE Conf. on Decision and Control*, San Diego, 1997.
- [45] R. Brown, e P. Hwang, *Introduction to Randon Signals and Applied Kalman Filtering*. 3rd edição, Nova York : John Wiley & Sons, 1997.
- [46] R. C. Arkin, *Behavior-based robotics*. Reading, MA: The MIT Press, 1998.
- [47] R. C. Arkin, e T. Balch, "AuRA: principles and practice in review," *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2), pp. 175-189, 1997.
- [48] R. Carelli, H. Secchi e V. Mut. "Algorithms for Stable Control of Mobile Robots with Obstacle Avoidance," *Latin American Applied Research*, pp.191-196, 1999.
- [49] R. Carelli, H. Secchi, V. Mut e O. Nasisi. "Algoritmos Estables para Navegación de un AGV en Pasillos Usando Flujo Óptico," *Anales de la VIII Reunión de Trabajo en Procesamiento de la Información y Control, RPIC'99*, Mar del Plata, Argentina, 23-25, 1999, pp. 79-7 a 86-7.
- [50] R. F. Vassallo, H. J. Schneebeli, J. Santos-Victor, "Visual navigation: combining visual servoing and appearance based methods," *SIRS'98, Int. Symp. on Intelligent Robotic Systems*, Edinburgo, Escócia, 1998.
- [51] R. H. Ruspini, A. Saffiotti e K. Konolige, "Progress in Research on Autonomous Vehicle Motion Planning", In: J. Yen, R. Langari, "*Industrial Applications of Fuzzy Logic and Intelligent Systems*", IEEE Press, Piscataway, NJ, 1995, Capítulo 8.
- [52] R. Kelly e R. Carelli, "A class of nonlinear PD-type controllers for robot manipulators." *Journal of Robotic Systems*, 13 (12), pp. 793-802, 1996.
- [53] T. F. Bastos, M. Sarcinelli, E. O. Freire, R. A. C. Freitas, H. A. Schneebeli, "Object Recognition for an Agent-Based Controlled Mobile Robot," *Proceedings of 5th IFAC Workshop on Intelligent Manufacturing Systems*, Gramado-RS, 9-11 de Novembro de 1998, pp. 323-328, 1998.
- [54] T. F. Bastos, Seguimiento y análisis de entornos de soldadura por arco automatizado mediante ultrasonidos. Tese de Doutorado, Universidad Complutense de Madrid, Espanha, 1994.
- [55] V. Chankong e Y. Haimes, *Multiobjective decision-making – theory and methodology*. Vol. 8, North-Holland, Amsterdam, Holanda, 1983.
- [56] V. Mut, R. Carelli e B. Kuchen, "Control de Impedancia Adaptativo para el Control de Robots con Realimentación Sensorial," *XIII National Symposium of Automatic Control - AADECA*, Argentina, 1992.

- [57] Zhi-Fang Yang e Wen-Hsiang Tsai. "Viewing corridors as right parallelepipeds for vision-based vehicle localization," *IEEE Trans. on Industrial Electronics*, vol. 46, No.3, Junho de 1999.

### **Bibliografia Consultada**

- [58] A. Elfes, "Sonar-based real-world mapping navigation," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3, pp. 149-165, Junho de 1987.
- [59] A. Kak e A. Kosaka, "Multisensor Fusion for Sensory Intelligence in Robotics," *Proceedings of Workshop on Foundations of Information/Decision Fusion: Applications to Engeneering Problems*, Washington D.C., EUA, Agosto de 1996.
- [60] A. Ram, R. C. Arkin, K. Moorman, e R. J. Clark, "Case-based reactive navigation: A case-based method for on-line selection and adaptation of reactive control parameters in autonomous robotic systems" *Technical Report GIT-CC-92/57*, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, 1992.
- [61] A. Saffiotti "Handling uncertainty in control of autonomous robots," *Uncertainty in Information Systems, Lecture Notes in Computer Science*, Springer-Verlag, 1998.
- [62] C. Ye and D. W. Wang, "A novel navigation method for autonomous mobile vehicles," *Journal of Intelligent and Robotic Systems*, Vol. 32, No. 4, pp. 361-388, 2001.
- [63] D. Langer, J. K. Rosenblatt e M Hebert, "A Behavior-Based System for Off-Road Navigation", *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 6, pp. 976-983, Dezembro de 1994.
- [64] D. Langer, J.K. Rosenblatt e M. Hebert, "An Integrated System For Off-Road Navigation," *Proceedings of IEEE Conference on Robotics and Automation*, San Diego, CA, 1994.
- [65] D. Payton, J. Rosenblatt e D. Keirsey, "Plan Guided Reaction," *IEEE Transactions Systems, Man, and Cybernetics*, Vol. 20, No. 6, pp. 1370-1382, Novembro de 1990.
- [66] E. Bicho e G. Schöner, "The dynamic approach to autonomous robotics demonstrated on a low-level vehicle plataform," *Robotics and Autonomous Systems*, No. 21, pp. 23-35, 1997.
- [67] E. Bicho, P. Mallet e G. Schöner, "Target Representation on an Autonomous Vehicle with Low-Level Sensors," *The International Journal of Robotics Research*, Vol. 19, No. 5, Maio de 2000, pp. 424-447.

- [68] E. Bicho, P. Mallet e G. Schöner, "Using attractor dynamics to control autonomous vehicle motion," *IECON'98 24<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society*, Aachen, Alemanha, 31 de Agosto a 4 de Setembro de 1998, Vol. 2, pp. 1176-1181.
- [69] E. Coste-Manière, H. H. Wang e A. Peuch, "Control architectures : what's going on?" *Proceedings of the International Program Development in Uric*, Lisboa, 1995. pp. 54-60.
- [70] E. Gat, R. Desai, R. Ivlev, J. Loch e D. P. Miller, "Behavior Control for Robotic Exploration of Planetary Surfaces," *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 4, pp. 490-503, Agosto de 1994.
- [71] H. Blaasvaer, P. Pirjanian e H. I. Christensen, "AMOR: An Autonomous Mobile Robot Navigation System," *IEEE International Conference on Systems, Man and Cybernetics* Vol. 3, Outubro de 1994, pp. 2266-2271.
- [72] H. R. Beom e H. S. Cho, "A sensor-based navigation for a mobile robot using Fuzzy Logic and Reinforcement Learning", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 3, pp. 464-477, 1995.
- [73] H. Ye, A. N. Michel e L. Hou, "Stability Theory for Hybrid Dynamical Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, pp. 461-490, Abril de 1998.
- [74] J. Borenstein e L. Feng, "Correction of Systematic Odometry Errors in Mobile Robots," *Proceedings of the 1995 International Conference on Intelligent Robots and Systems (IROS'95)*, Pittsburg, Pennsylvania, Agosto de 1995, pp. 569-574.
- [75] J. Borenstein e L. Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, Outubro de 1996.
- [76] J. K. Rosenblatt e C. Thorpe, "Combining Multiple Goals in a Behavior-Based Architecture," *Proceedings of 1995 International Conference on Intelligent Robots and Systems (IROS)*, Pittsburgh, PA, de 7 a 9 de Agosto de 1995.
- [77] J. K. Rosenblatt, "DAMN: A Distributed Architecture for Mobile Navigation," *Proceedings of 1995 AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents*, Stanford , CA, 27 a 29 de Março de 1995.
- [78] J. Richardson e K. Marsh, "Fusion of Multisensor Data," *The International Journal of Robotics Research*, Vol. 7, No. 6, pp. 78-96, Dezembro de 1988.
- [79] J. Yen e N. Pfluger, "A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 6, pp. 971-978, Junho de 1995.

- [80] L. Basañez, "Multi-sensor integration in robotics," *1<sup>o</sup> Workshop on Robotics and CIM*, Lisboa, 1989, pp. 1-34.
- [81] L. Hong, "Sense Your World Better: Multisensor/Information Fusion," *IEEE Journal of Circuits and Systems*, Vol. 10, No. 3, pp. 7-8;12-15;28, 1999.
- [82] M. Egerstedt e X. Hu, "A hybrid control approach to action coordination for mobile robots," *Automatica*, No. 38, pp. 125-130, 2002.
- [83] M. Khatib, B. Bouilly, T. Siméon e R. Chatila, "Indor Navigation with Uncertainty using Sensor-Based Motions," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, Novo México, Abril de 1997, pp. 3379-3384.
- [84] M. Mataric, "A distributed model for mobile robot environment-learning and navigation," *Technical Report AI-TR-1228*, Massachusetts: MIT Artificial Intelligence Laboratory, 1992.
- [85] N. C. Tsourveloudis, K. P. Valavanis e T. Hebert, "Autonomous Vehicle Navigation Utilizing Eletrostatic Potential Fields and Fuzzy Logic", *IEEE Transactions on Robotics and Automation*, Vol. 17, No 4, Agosto de 2001.
- [86] P. Pirjanian "The Notion of Optimality in Behavior-Based Robotics", *Journal of Robotics and Autonomous Systems*, Special Issue, 1999.
- [87] P. Pirjanian e M. Mataric, "A decision-theoretic approach to fuzzy behavior coordination," *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Novembro de 1999.
- [88] P. Pirjanian e M. Mataric, "Multiple Objective vs. Fuzzy Behavior Coordination," *Lecture Notes in Computer Science on Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, 2000.
- [89] P. Pirjanian, "An Overview of System Architectures for Action Selection in Mobile Robotics," *Technical Report*, Laboratory of Image Analysis, Aalborg University, 1997.
- [90] P. Pirjanian, J. A. Fayman e H. I. Christensen, "Improving Task Reliability by Fusion of Redundant Homogeneous Modules Using Voting Schemes," *IEEE International Conference on Robotics & Automation*, Albuquerque, New Mexico, April 1997. pp 425-430.
- [91] R. A. Decarlo, M. S. Branicky, S. Pettersson e B. Lennartson, "Perspectives and Results on the Stability and Stabilizability of Hybrid Systems," *Proceedings of the IEEE*, Vol. 88, No. 7, pp. 1069-1082, Julho de 2000.

- [92] R. C. Arkin, "Towards the Unification of Navigation Planning and Reactive Control," *Working Notes of the AAAI Spring Symposium on Robot Navigation*, Stanford University, 28 e 29 de Março de 1989.
- [93] Z. Li, C. B. Soh e X. Xu, "Lyapunov stability of a class of hybrid dynamic systems," *Automatica*, No. 36, pp. 297-302, 2000.